

ЭКОСИСТЕМА АВИАЦИОННОЙ РАЗРАБОТКИ и СЕРТИФИКАЦИИ



**ПРИМЕНЕНИЕ
КТ-178С, КТ-254, Р4754А**

ВЭНС ХИЛДЕРМАН

С более чем 25 экспертами отрасли

Примечание переводчика

Русский перевод выполнен ChatGPT 5.5 за 243 итерации.

Авторское право © 2021, Вэнс Хилдерман

Все права защищены.

Aviation Press International, абонентский ящик 811547

Лос-Анджелес, Калифорния, США 90081

Содержание

Предисловие	3
Об авторе	5
Обзор автора и благодарности	7
Введение	12
Авиация и авионика: структура разработки и сертификации	15
Сертификаты типа (TC), технические стандартные приказы (TSO) и одобрение изготовителя частей (PMA)	23
ARP4761A и безопасность в авиации	42
ARP4754A: разработка воздушных судов и систем	67
DO-178C: вводная глава о ПО авионики	97
Авиационное ПО по DO-178C	107
Аппаратура авионики по DO-254	131
Авиационные требования к системам, ПО и аппаратуре	152
DO-200B: аэронавигационные данные	170
DO-278A: наземные и спутниковые системы и ПО связи, навигации, наблюдения и организации воздушного движения (CNS/ATM)	185
DO-331: разработка на основе моделей (MBD)	200
DO-332. Объектно-ориентированная технология (Object-Oriented Technology, OOT)	217
DO-330. Квалификация инструмента (TQ)	233
Авиационная кибербезопасность DO-326A	252
Инженерные переходы	278
Трассируемость в авиационной разработке	287
Валидация и верификация авиационных систем (V&V)	294
Испытания на внешние воздействия по DO-160	320
Гарантия качества и гарантия процесса в авиации	331
Соответствие требованиям в военной авиации	342
Многоядерные процессоры для авионики по CAST-32A (с интегрированной модульной авионикой (IMA) по DO-297)	355
Затраты и выгоды авиационной разработки и сертификации	377
Аспекты сертификации беспилотных летательных аппаратов (UAV) и беспилотных авиационных систем (UAS)	391
Заключение	401
Приложение А: Сокращения в авионике	404
Приложение В: Контрольный перечень требований к авиационной системе	427

Предисловие

Вудроу Беллами III, главный редактор Access Intelligence: Aerospace

«Способность Вэнса Хилдермана объяснять весь жизненный цикл: планирование, разработку, валидацию, верификацию и, наконец, сертификацию бортовых систем ПО и аппаратуры, критичных для безопасности, хорошо видна в его последней работе: «Экосистема авиационной разработки и сертификации» (Aviation Development & Certification Ecosystem)».

Одна фраза, сказанная мне г-ном Хилдерманом в интервью несколько лет назад, до сих пор служит ориентиром в моей работе и объясняет, как этот сегмент авиации будет развиваться дальше: как взаимосвязанная экосистема производителей самолетов, поставщиков авионики, организаций по обслуживанию воздушных судов, эксплуатантов и других участников.

«Все хотят подняться выше по пищевой цепочке. Производители компонентов хотят выпускать подсистемы, производители подсистем хотят выпускать полноценные системы, а разработчики систем хотят создавать интегрированные системы. Это невероятно динамичная среда».

Способность Вэнса Хилдермана объяснять весь жизненный цикл: планирование, разработку, валидацию, верификацию и, наконец, сертификацию бортовых систем ПО и аппаратуры, критичных для безопасности, хорошо видна в его последней работе. **«Экосистема авиационной разработки и сертификации»** действительно стала итоговой работой его 40-летней карьеры, в течение которой он обучал инженеров, впоследствии продвигавших развитие бортовой аппаратуры и ПО.

Вниманию начинающих инженеров, читающих эту книгу: впереди множество новых разработок, для которых нужно будет получить сертификаты типа, технические стандартные приказы (Technical Standard Order, TSO) и одобрения изготовителя частей (Parts Manufacturer Approval, PMA). Завтрашние электрические аэротакси и турбовинтовые самолеты на водородной тяге потребуют совершенно новых способов измерять мощность, показывать ее пилотам на борту и дистанционно управлять такими аппаратами, а также новых графических интерфейсов «человек-машина». Данные о местоположении таких аппаратов должны будут отслеживаться как традиционно пилотируемыми воздушными судами, так и системами организации воздушного движения (Air Traffic Management, ATM). И это только в рамках быстро растущего сегмента электрических аппаратов вертикального взлета и посадки (electric vertical take-off and landing, eVTOL).

Искусственный интеллект (Artificial Intelligence, AI) и машинное обучение (Machine Learning, ML) в наши дни для авиационного инженерного сообщества уже не просто модные слова. В феврале прошлого года Агентство Европейского союза по авиационной

безопасности (European Union Aviation Safety Agency, EASA) прямо сообщило мне, что ожидает к 2025 году сертификации первой функции системы управления полетом воздушного судна транспортной категории (Flight Control System, FCS), использующей искусственный интеллект. Однако, когда я слышу или вижу пресс-релизы о впечатляющих, почти фантастических концепциях воздушных судов, я стараюсь помнить одну вещь:

Сегодня самые массовые воздушные суда, находящиеся в эксплуатации у крупнейших коммерческих авиакомпаний, оснащены системами управления полетом, которые должны соответствовать следующему основополагающему принципу:

Для воздушных судов, перевозящих людей, вероятность катастрофического отказного состояния, вызванного отказом системы воздушного судна, должна быть ниже 10^{-9} .

Это означает, как и всегда означало для инженеров по авиационной электронике, что с каждой строкой кода, с каждой операционной системой реального времени (Real-Time Operating System, RTOS), с каждой передачей по шине данных и с каждым процессом гарантии качества (Quality Assurance Process, QA) при разработке новых авиационных технологий связана чья-то жизнь. По мере того как отрасль в 2020-х годах продолжает осваивать невиданные прежде инженерные решения, используйте эту книгу, *«Экосистема авиационной разработки и сертификации»*, как руководство по созданию будущего.

В течение последних девяти лет я перевожу технический жаргон, описывающий достижения в области авиационной электроники, в понятные и практически полезные материалы: статьи и другой контент для Avionics International, входящего в Access Intelligence LLC.

Вудроу Беллами III

Об авторе



Вэнс Хилдерман – эксперт с 35-летним опытом в области критичных для безопасности авиационных и авионических разработок. Он получил степень бакалавра наук в области электротехники (BSEE, Bachelor of Science in Electrical Engineering) и степень магистра делового администрирования (MBA, Master of Business Administration) в Университете Гонзага, а также степень магистра компьютерной инженерии в Университете Южной Калифорнии (USC, University of Southern California), где был стипендиатом Hughes. На протяжении всей карьеры г-н Хилдерман занимается критичным для безопасности авиационным и авионическим ПО, безопасностью, разработкой систем и аппаратуры, а также связанными с этим техническими решениями для сертификации. В 1990 году г-н Хилдерман совместно с г-ном Биллом Хаббардом основал TekSci. Компания стала крупнейшим в мире независимым поставщиком услуг в области авионического ПО и сертификации, где работало более 150 штатных инженеров. Все четырнадцать лет существования TekSci он единолично занимал должности генерального директора (CEO, Chief Executive Officer), президента и главного технического директора (CTO, Chief Technical Officer). Г-н Хилдерман разработал первые в мире учебные курсы по DO-178, DO-200, DO-278 и DO-254, которые приобрели инженеры из 90 % авиационных компаний мира. Он также провел более 170 анализов пробелов (Gap Analysis) по документам серии DO-XXX и ARP4754A/4761 для более чем восьмидесяти компаний, работая на площадках в тридцати странах. Г-н Хилдерман консультировал 70 % из 200 крупнейших в мире авиационных компаний-разработчиков, а решения, созданные при его участии, эксплуатируются в большинстве воздушных судов и систем, на которые ежедневно полагаются по всему миру. Он также разработал передовые технологии и решения для ARP4761, операционных систем реального времени, OCPB (Real-Time Operating System, RTOS), и многоядерных вычислений в соответствии с документом CAST-32A Группы сертификационных органов по ПО (Certification Authorities Software Team, CAST), Федерального управления гражданской авиации США (Federal Aviation Administration,

FAA) и Агентства Европейского союза по авиационной безопасности (European Union Aviation Safety Agency, EASA). Государственные органы разных стран обращаются к г-ну Хилдерману за выездными услугами по авиационной сертификации, включая аудит, анализ пробелов, наставничество, проектирование и обучение персонала в США, Канаде, Германии, Бразилии, Австралии, Новой Зеландии, Италии, Великобритании, Южной Корее, Испании, Израиле и Турции.

В 2005 году г-н Хилдерман стал основным учредителем HighRely Inc. и на протяжении всего периода существования компании единолично занимал должности генерального директора, президента и главного технического директора. HighRely занималась сертификацией авиационного ПО и систем; в 2011 году ее приобрела Atego/Artisan, при этом г-н Хилдерман до 2013 года оставался главным техническим директором по авиационным и сертификационным услугам и сохранил все авторские права на свои предыдущие работы в соответствии с законодательством штата Калифорния об авторских правах. Г-н Хилдерман также разработал первый подход к сертификационному анализу пробелов для DO-178, DO-278A, DO-200, ARP4754A и DO-254 и был наставником для более чем 75 компаний, создающих критичные для безопасности изделия, помогая им выстраивать экономически эффективную разработку авионических проектов.

С технической точки зрения г-н Хилдерман – системный инженер, а также инженер по ПО и вычислительной технике. Он специализируется на разработке системных и программных требований, проектных решений, кода и тестов. Кроме того, он является экспертом по стратегиям сертификации бортовых и наземных авиационных систем связи, навигации, наблюдения и организации воздушного движения (Communication, Navigation, Surveillance / Air Traffic Management, CNS/ATM). Г-н Хилдерман работал научным сотрудником в Hughes Aircraft, где занимался наземными системами наблюдения и радиолокации с бортовыми интерфейсами. Хилдерман – основной автор самой продаваемой в мире книги по авионической сертификации, в частности по DO-178C и DO-254, а также автор более 30 технических статей о разработке и сертификации авиационных систем и ПО. Ряд профессиональных организаций выбрал его своим единственным преподавателем по DO-178C, DO-254 и ARP4754A, включая Aviation Electronics Europe, Aerospace Tech Conferences, Конференцию по цифровым авиационным системам (Digital Avionics Systems Conference, DASC) и SAE (издатель стандартов). Хилдерман лично обучил более 25 000 авиационных инженеров по всему миру и по-прежнему ежегодно обучает более 2 000 инженеров по документам серии DO-XXX и ARP4754A, одновременно консультируя более 40 авиационных компаний в год.

Г-н Хилдерман считает своими сильными сторонами честность, умение выстраивать отношения, способность доносить сложные решения и проактивное профессиональное лидерство, которое помогает мотивировать инженеров его компании AFuzion на максимальные результаты. Все это сочетается с постоянными техническими инновациями. Безопасного неба.

Обзор автора и благодарности

Написать эту книгу казалось отличной идеей... тогда. Как и у многих на первый взгляд “отличных идей”, реализация оказалась сложнее, чем представлялось. И, как выяснилось, были причины, по которым более способные люди не довели до конца похожий замысел: написать одну краткую книгу с самой важной информацией, которая уже сегодня нужна разработчикам авиации, воздушных судов и авионики завтрашнего дня. Настоящая трудность состояла в том, чтобы сжать сорок лет опыта и тысячи страниц заметок в краткую, удобочитаемую книгу без пустословия и проповеднического тона. К счастью, тридцать лучших опытных авиационных инженеров, с которыми мне довелось работать, помогли, надеюсь, достичь этой цели: создать книгу, которую вы сейчас читаете.

Хотя спор о том, кто первым поднялся в воздух, продолжается (подсказка: у американцев и у наших французских коллег могут быть разные мнения об эффективности стартов с направляющего рельса в Китти-Хоке), все согласны в одном: авиационные технологии жизненно важны, а их развитие впечатляет. Будущие авиационные системы на базе электрического аппарата вертикального взлета и посадки (electric vertical take-off and landing, eVTOL), городской воздушной мобильности (Urban Air Mobility, UAM) и искусственного интеллекта (Artificial Intelligence, AI) показывают, насколько далеко продвинулась отрасль уже сегодня. Однако за экспоненциально растущей сложностью авиационных систем скрывается столь же растущая сложность разработки и сертификации. Хотя безопасность авиации в целом повышалась, недавние трагедии с несколькими катастрофами показывают, что такую сложность нелегко понимать и контролировать. Авиационные проекты требуют одновременной работы десятков, сотен, а для сложной бортовой или наземной системы – тысяч инженеров. Чтобы создать безопасное воздушное судно, действительно нужны более 5000 инженеров, но достаточно всего нескольких человек, чтобы пропустить фатальные ошибки. Единственное решение предполагает наличие “Экосистемы авиационной разработки” (Aviation Development Ecosystem), которая дает каждому участнику понятный набор правил и ориентиров для применения на практике. Однако официального единого справочника по Экосистеме авиационной разработки не существует, а многие ведомства и группы в разной степени участвуют в определении ее отдельных частей. Так и родилась моя, казавшаяся отличной, идея этой книги: систематизировать Экосистему авиационной разработки, применимую во всем мире.

Как разработчик авионики и один из главных основателей трех наиболее значимых за последние четыре десятилетия компаний, оказывающих услуги по разработке и сертификации авионики и авиационной техники, я почти не помню недели, чтобы какой-нибудь клиент в области авиационной разработки не спрашивал меня, где найти единый источник информации, собранной в этой книге. Когда я объяснял, что найти такой источник столь же невозможно, как источник вечной молодости, обычный ответ был

таким: “Не верю. В наши дни, когда ко всему цифровому есть простой доступ, наверняка должен существовать какой-то источник информации! Я готов заплатить – где можно купить книгу, в которой все это собрано?” Молчание...

Действительно, после нескольких лет поисков настоящего источника вечной молодости бесстрашный, хотя и чрезмерно оптимистичный, исследователь Понсе де Леон, думаю, тоже был бы готов заплатить за точное местонахождение этого источника. Но даже сегодня мы все еще находимся в начале цифровой эпохи. Я не сомневаюсь, что через одно-два десятилетия искусственный интеллект и облачные технологии разовьются настолько, что позволят быстро искать по всему известному цифровому миру и получать индивидуально сформированную “книгу”, разъясняющую все темы, которые затронуты здесь. Но будет ли эта информация выверенной и точной? И будет ли она представлена в формате, который позволит читателю по-настоящему понять, что, почему и как делается при разработке и сертификации авиационных систем?

Ни одна книга не способна удовлетворить всех и во всем: люди слишком различаются по знаниям, потребностям и возможностям. Поэтому, как автор, который разрабатывал, собирал, запрашивал, отбирал, компилировал и писал представленную здесь информацию, я руководствовался “правилом 80 %”. Его я привожу в начале своих учебных курсов по авиационной разработке, которые прошли более 25 000 слушателей по всему миру: “Если 80 % этой информации на 80 % полезны для 80 % слушателей, значит, как преподаватель я добился 100 % успеха”. В идеальном случае? Надеюсь. В реальной жизни? Безусловно.

Эта книга действительно стала результатом тысяч часов напряженной работы многих людей. Иногда захватывающей, гораздо чаще – не настолько захватывающей. Но всегда интересной. Совокупный опыт участников и рецензентов превышает пятьсот человеко-лет, а это немало, особенно если помнить, что самой коммерческой авиационной отрасли немногим более ста лет. Каждый из многих людей, участвовавших в создании этой книги, заслуживает искренней благодарности. А читателям я скажу так: если вы просто приложите свои лучшие усилия к тому, чтобы сделать авиацию хоть немного лучше, немного предсказуемее и немного безопаснее, для меня это будет ценнее любой формальной благодарности. Если у вас есть комментарии или рекомендации для будущих изданий, вы всегда можете связаться со мной: поиск моих контактных данных займет меньше минуты, но спамерам мы все же немного усложним задачу.

Повторю: эта книга была бы невозможна без прямого и косвенного вклада столь многих людей. Сердечно благодарю мою семью, которой пришлось провести немало вечеров и выходных за обсуждением идей о содержании и контексте. Большое спасибо Emilie Mandic и Anna Hilderman, помогавшим редактировать ранние версии этих глав. Большое спасибо всем техническим участникам и рецензентам, чей опыт отражен на этих страницах. Огромное спасибо более чем 300 моим клиентам и техническим коллегам, которые неформально рецензировали ранние главы и давали обратную связь. И еще одна

особая благодарность официальным рецензентам глав, а в некоторых случаях и авторам, перечисленным ниже: почти все ваши сотни комментариев и дополнений были учтены, и благодаря вам эта книга стала намного лучше.

Еще раз хочу лично отметить и поблагодарить многих официальных экспертных технических рецензентов и участников, задействованных в этой работе: на таких людях держится современная авиация, и благодаря им мир стал лучше и безопаснее. Огромное спасибо моей прекрасной жене Colleen Hilderman, чья внешняя красота уступает только внутренней: ты вдохновляешь всех нас. Спасибо и четверым из наших шестерых детей (Anna, Emilie, Evan и Dan), которые помогали редактировать или рецензировать части этой книги: вас ценят и любят, как и двоих других (Carson и Kaitlin), которые каким-то чудом оказались заняты чем-то другим именно тогда, когда потребовалось раннее рецензирование.

Всем перечисленным выше: спасибо за дополнительное время, благодаря которому читатели смогут воспользоваться вашими многочисленными замечаниями и предложениями. Кратко и ясно разъяснить сложные темы совсем не просто. Куда легче было бы написать еще одну тысячестраничную техническую тираду, которую у реально работающих специалистов нет ни времени, ни желания читать.

Безопасного неба,

Вэнс Хилдерман

Генеральный директор (Chief Executive Officer, CEO) и технический директор (Chief Technology Officer, CTO), AFuzion Inc., www.afuzion.com

- Бакалавр электротехники, *Университет Гонзага (Gonzaga University)*
- Магистр электротехники и вычислительной техники, *Университет Южной Калифорнии (University of Southern California, USC)*
- Магистр делового администрирования (Master of Business Administration, MBA), *Университет Гонзага (Gonzaga University)*

Особая **БЛАГОДАРНОСТЬ** официальным рецензентам глав и участникам, перечисленным ниже:

Технические участники, в алфавитном порядке:

1. Г-н Mark Brown, старший архитектор, Lynx Software
2. Г-н Aharon David, эксперт по кибербезопасности и авиации, AFuzionInfosec
3. Г-н Jon Lynch, представитель Федерального управления гражданской авиации США (Federal Aviation Administration, FAA) по инженерной экспертизе (Designated Engineering Representative, DER), AFuzion
4. Г-н Greg Wilson, представитель FAA по инженерной экспертизе

Технические рецензенты, в алфавитном порядке:

1. Г-н Brian Allen, ведущий инженер по ПО, Sikorsky, компания группы Lockheed Martin
2. Г-н Don Coleman, представитель FAA по инженерной экспертизе
3. Г-н Bunyamin Coskun, руководитель проекта, Milsoft Inc.
4. Г-н Kevin Crozier, старший инженер по ПО авионики, консультант, представитель FAA по инженерной экспертизе, летный инструктор, сертифицированный FAA, и коммерческий пилот.
5. Г-н Bill De Leeuw, назначенный представитель FAA по системам, оборудованию и ПО
6. Dave Deloria, директор по инженерным разработкам, Crane Aero.
7. Г-н Eric Dillaber, руководитель разработки направления сертификации и стандартов Simulink
8. Д-р Bruce Douglass, главный евангелист направления Интернета вещей (Internet of Things, IoT) в IBM
9. Г-жа Gamze Eroglu, бывший старший специалист по сертификации, STM
10. Г-н Marc GATTI, научный директор Thales AVS France SAS
11. Г-жа Nazan Gozay Gurbuz, эксперт Taos по безопасности и член комитета ARP4754A/4761A
12. Г-н Francois Guay, представитель FAA по инженерной экспертизе
13. Г-н Alan Houp, независимый эксперт по DO-160
14. Г-н Kevin Kendryna, руководитель по качеству инженерных разработок, General Atomics
15. Д-р J. Lee, руководитель направления авионики и полковник Военно-воздушных сил Республики Корея в отставке
16. Г-н Russell McMullan (бывший технический директор BECA)
17. Г-н George Meier, представитель FAA по инженерной экспертизе
18. Г-н Vincenzo Mirra, менеджер по качеству и сертификации в аэрокосмической отрасли, Leonardo Company
19. Г-н Roger Plieske, Rohde & Schwarz
20. Г-н Bill Potter, менеджер по техническому маркетингу, The MathWorks
21. Г-н Tom Roth, представитель FAA по инженерной экспертизе

22. Г-н Antonio Silva (магистр наук по электротехнике (Master of Science in Electrical Engineering, MSEE), профессиональный инженер (Professional Engineer, PE)), руководитель программной инженерии в AgustaWestland, инженер по верификации соответствия ПО и бортовой электронной аппаратуры (Airborne Electronic Hardware, АЕН)
23. Г-н Carmelo Tommasi, эксперт по моделированию ПО, AFuzion
24. Г-жа Lisa Wynn (Duncan), инженер по качеству ПО, магистр искусств (Master of Arts, M.A.), сертифицированный инженер по качеству ПО (Certified Software Quality Engineer, CSQE).

Введение

«Это была тёмная и бурная ночь...»

Постойте. Эта книга о том, как разобраться в области авиационной разработки и в множестве связанных с ней «стандартов». Что здесь делает в качестве первой фразы клише 1830 года, появившееся задолго до авиации, – про «тёмную и бурную ночь»? Всё просто: если вы, как и я, проходили лётную подготовку, то знаете, что тёмные и бурные ночи редко бывают вам другом. То же верно для разработчиков критически важных авиационных систем и систем авионики: тёмная кабина и неожиданные сценарии непогоды никогда не играют на вашей стороне. Но в авиации, как и в жизни, мы можем выбирать хороших друзей, хотя полностью уйти от потенциально вредных влияний не можем. Разницу между хорошим и плохим исходом определяет наш выбор. А для хорошего выбора нужны серьёзные знания. И опыт. В авиации опыт ничем не заменить: он вам нужен, и пока его нет, вы в лучшем случае остаётесь «вторым пилотом».

К счастью, и пилотам, и авиационным разработчикам доступны «друзья», которые уменьшают влияние тёмных бурь. Что это за друзья? Воздушные суда оснащаются всё более совершенными бортовыми системами авионики и наземными системами, а это оборудование разрабатывается с использованием руководящих материалов, разбираемых в этой книге. Вместе они рассеивают тьму и укрощают бури. Легко, верно? Совсем нет: чтобы друзья помогали, они должны быть под рукой, а вы должны знать, как с ними работать.

Однако даже при хороших друзьях по разработке остаётся одна сложность. Именно «сложность», а не «проблема», потому что сложность – это проблема, у которой есть решение. Быстро: какое решение? Ага, мы забегаем вперёд. Сегодня, когда доступ к цифровым средствам и умение ими пользоваться стремительно развиваются, повседневную работу всё чаще пронизывает желание найти быстрое решение, не разобравшись до конца в полной экосистеме проблемы. Но сначала надо понять: в чём сложность? И это тоже непростой вопрос, потому что в тёмной и бурной ночи разработки авиационных систем таких сложностей несколько, в том числе следующие:

Каждая авиационная система отличается от всех остальных систем

- *Руководящие материалы – это именно руководящие материалы: общие указания, которым не хватает понятных целей, ясности и методики*
- *Не существует одного оптимального подхода, подходящего всем*
- *Не существует внешнего рецепта успеха, поэтому срывы по качеству, стоимости и срокам встречаются часто.*

В этой непростой авиационной среде, где сложностей много, нам нужно быстро найти наших друзей. Нам нужно разобраться во множестве авиационных руководящих материалов, стандартов, меморандумов и политик. Каждый авиационный инженер должен без труда понимать, как экосистема авиационной разработки (aviation development ecosystem) применяется к его конкретной ситуации: к авиационным системам, воздушным судам, аппаратуре и ПО, которые он разрабатывает.

Можно с большой уверенностью предположить, что большинство читателей этой книги проходили курсы высшей математики, где в начале обучения возникала похожая растерянность. Помните эти времена? Студенты не хотели поднимать руку и задавать вопрос, потому что им казалось, будто все остальные уже всё поняли; каждый думал, что трудно только ему. Потом понимаешь, что остальные были такими же: у них были те же учебники, и они тоже оставались в неведении. Как специалист по авионике и авиации с тридцатипятилетним опытом консультирования более чем по 500 авиационным проектам в 30 странах, я лично заверяю вас: сложности есть у всех; большинство считает, что у них проблемы, а идеальные решения есть у очень немногих проблем.

Возможно, читатель умнее своих коллег (в конце концов, он приобрёл эту книгу), но сложности у него всё равно были. И он достаточно умён, чтобы понимать: не существует единого Розеттского камня, с помощью которого можно расшифровать код разработки проектов с минимальными затратами и сроками и одновременно получить максимальное качество и безопасность. Бесплатных обедов не бывает, даже в государственных или корпоративных программах бесплатного питания: кто-то всё равно заплатил. Но почти у каждого есть конкуренты, и у этих конкурентов те же сложности, а возможно, даже и «проблемы». В этой книге я надеюсь использовать знания, накопленные почти за четыре десятилетия опыта, и передать полезную часть знаний, которыми я лично делился, обучая более 25 000 авиационных инженеров и более 500 сотрудников тому, как добиваться инженерного и сертификационного успеха. Эта книга даёт знания, которые помогут вам стать лучшим вторым пилотом и в конечном счёте хорошим пилотом собственного авиационного успеха... возможно, даже отличным пилотом.

Что делает техническую книгу «хорошей»? Это почти невозможный вопрос, потому что ответ зависит от того, кто смотрит. У каждого читателя этой книги свои потребности, свои вопросы и свой способ учиться. Действительно, невозможно быть всем для всех. Но если 80 % читателей сочтут, что 80 % собранных здесь знаний полезны им на 80 %, я скажу, что результат получился настолько хорошим, насколько это вообще возможно. Разумеется, было бы легко просто пересказать опубликованные руководящие материалы: например, документ «Аспекты ПО при сертификации бортовых систем и оборудования» (DO-178C, Software Considerations in Airborne Systems and Equipment Certification), документ «Руководство по обеспечению целостности ПО систем связи, навигации, наблюдения и организации воздушного движения» (DO-278A, Guidelines for Communication, Navigation, Surveillance, and Air Traffic Management Systems Software Integrity Assurance), руководство по разработке гражданских воздушных судов и систем (ARP-4754A, Guidelines for Development of Civil Aircraft and Systems) и т. д. Или

пересказать разъяснительный материал «Вспомогательная информация к DO-178C и DO-278A» (DO-248C, Supporting Information for DO-178C and DO-278A), который, собственно, должен объяснять документ DO-178C. Но вы, вероятно, уже просматривали эти документы и увидели, что их недостаточно. Почему? Это руководящие материалы общего назначения, предназначенные для кодификации указаний по разработке авиационных систем и систем авионики. Не хватает объяснений простым языком и сведения всей авиационной **экосистемы** в единую картину, актуальную на протяжении всего инженерного жизненного цикла и изложенную в контексте реального мира и его лучших практик. Эта экосистема включает взаимосвязь и переход от безопасности воздушного судна и систем к безопасности ПО и аппаратуры. Не теоретические рассуждения из учебной аудитории, а практическое понимание реального мира, потому что это и есть ваш мир: реальный. Итак, запускаем двигатели и начинаем успешный учебный полёт...

Авиация и авионика: структура разработки и сертификации

Автор: Вэнс Хилдерман

Рецензенты:

г-н Francois Guay, представитель Федерального управления гражданской авиации США (Federal Aviation Administration, FAA) по инженерной экспертизе (Designated Engineering Representative, DER); г-н George Meier, представитель FAA по инженерной экспертизе (DER)

Разработка авиационной техники важна для тех, кому важна сама авиация. Кто эти люди? Разработчики, изготовители, интеграторы, летные экипажи, авиакомпания-эксплуатанты и просто энтузиасты авиации хорошо понимают ее значение. Кому стоит читать эту книгу? Ответ прост: каждому, кто участвует в экосистеме авиационной разработки. Пассажиры тоже согласятся, как только осознают, что речь идет об их безопасности.

Сотни лет назад, еще до того как замысел летательной машины да Винчи посеял семена будущих аэростатов, планеров и самолетов, людям поневоле приходилось быть мастерами на все руки. Среднему жителю Земли нужны были практические знания о погоде, земледелии, животноводстве, строительстве и деторождении. Со временем человечество пришло к специализации как к прагматичному решению: производительность росла, когда люди могли сосредоточиться на своих навыках. Так возникали гильдии, профессиональные союзы, государственные институты и нормативные требования, которые, к лучшему или к худшему, повышали качество жизни обычного человека.



Одним из результатов такой специализации стало отделение использования изделия от знания о его проектировании. Пользователю сельскохозяйственного орудия уже не нужно было знать, как это орудие спроектировать; он в определенной степени доверял изготовителю, исходя из отношений с ним и уплаченной цены. Так появились предпосылки современной модели доверия к технике... а затем появились современные самолеты.

Воздушные суда и вообще летательные объекты (аэростаты, ракеты и беспилотные авиационные системы) создают опасности гораздо более серьезные, чем сельскохозяйственные орудия. Сломанное орудие просто замедляет работу, пока его не починят. Отказавшее воздушное судно может иметь куда более тяжелые последствия: замедление жизни может стать окончательным. Поэтому государства и разработчики совместно начали закреплять безопасные основы разработки и сертификации авиационных компонентов и систем. Участники почти всегда руководствовались благими намерениями, но сталкивались с непростыми компромиссами. Перерегулировать отрасль – значит пожертвовать конкурентоспособностью и инновациями, что в итоге тоже снижает безопасность. Недорегулировать – значит поставить безопасность под угрозу с потенциально катастрофическими последствиями. Дилемма, что и говорить.

Безопасности хотят все хотя бы из эгоистических соображений: каждый либо летает, либо потенциально может оказаться под падающим воздушным судном. При этом авиационная отрасль относится к самым прагматичным и наименее эгоистичным отраслям из существующих (по крайней мере, по мнению автора, если сравнивать с банками, правительствами, юристами и политиками). Для авиации безопасность полезна и для здоровья, и для бизнеса. Ничто так не охлаждает интерес к полетам и коммерческие перспективы, как тяжелая авиационная катастрофа. Достаточно вспомнить Concorde: один из самых безопасных и красивых самолетов, пока первая в его истории катастрофа в 2000 году не привела к окончательному прекращению эксплуатации всего парка.

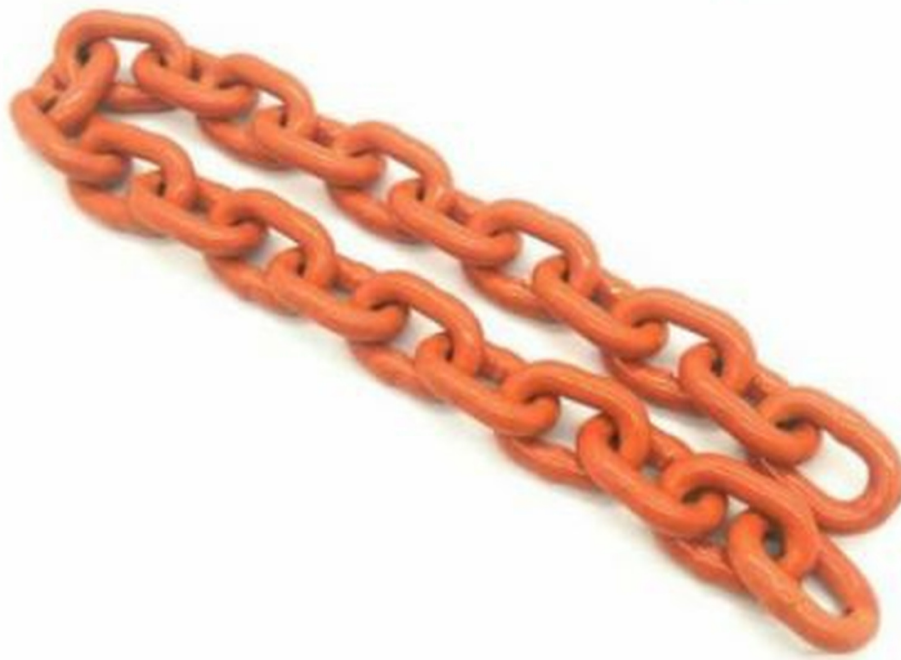
К счастью, многие умные и практичные люди десятилетиями формировали правила обеспечения безопасности полетов. В мире существуют международные механизмы и для других сфер человеческой деятельности, включая поддержание мира (Организация Объединенных Наций) и банковское дело (Всемирный банк и Международный валютный фонд, МВФ), но большинство согласится, что организации в области безопасности полетов относятся к числу наиболее эффективных, невзирая на стоимость. Система обеспечения безопасности полетов в целом хорошо финансируется, опирается на науку, содержит умеренную долю политики и, по крайней мере в западных странах, имеет разумное сотрудничество и поддержку со стороны авиационной отрасли. Это не значит, что все идеально; вовсе нет. Есть заметные случаи политиканства, национализма и использования влияния сильными участниками в собственных интересах. К спорным вопросам в авиации относятся выбросы углекислого газа (CO₂), правила работы пилотов, снижение шума, сборы за пролет, растущая загруженность воздушного движения, правила владения авиакомпаниями и гармонизация трансатлантического регулирования. Рост Азии как экономической силы и как участника авиации требует дополнительной координации для решения связанных с этим задач. Но в целом система обеспечения безопасности полетов остается работоспособной. Так что же это за система? На следующем рисунке показаны уровни безопасности в авиации:

- **Воздушное пространство** связано с планированием полетов, секторами и государственными границами, а также с одновременным управлением множеством воздушных судов и операций.
- **Связь, навигация, наблюдение и организация воздушного движения (OpВД; Communication, Navigation, Surveillance / Air Traffic Management, CNS/ATM)** охватывает наземные и спутниковые виды деятельности, относящиеся к воздушным судам и безопасности полетов.
- **Воздушное судно** охватывает аспекты безопасности, относящиеся к отдельному воздушному судну.
- **Авионическая система** охватывает аспекты безопасности конкретной системы авионики; воздушное судно состоит из множества таких систем.



Как видно на диаграмме, безопасность относится ко всей авиационной экосистеме, а не только к отдельным ее частям. Разработчики авионики часто любят считать себя вершиной пищевой цепочки. Но, как ясно показывает диаграмма, авионические системы находятся внизу. Почему? Потому что авиационная экосистема больше любой отдельной системы и даже любого отдельного воздушного судна.

Очевидно, одни части экосистемы безопасности полетов вносят в безопасность больший вклад, чем другие. Но если спросить эксперта по безопасности полетов: “Какой аспект безопасности в авиации самый важный?”, правильный ответ будет: “Все”. Иными словами, хотя некоторые аспекты могут сильнее влиять на безопасность или требовать большей строгости, важны все аспекты внутри этой экосистемы. Но современный человеческий мозг работает иначе: сегодня все должно быть ранжировано. Задумайтесь: ловили ли вы себя на мыслях о своем весе, росте, физической форме, часах сна, счете в гольфе, банковском балансе, калориях, связях в социальных сетях или зарплате? Поздравляю, это нормально. Но безопасность полетов связана с экосистемой, которая не поддается простому ранжированию, потому что каждый аспект вносит свой вклад в безопасность и потому важен сам по себе.



Еще один способ взглянуть на безопасность полетов – представить простую цепь. Она похожа на безопасность в авиации, потому что каждое звено участвует в функции цепи и в ее надежности. Хотите найти самое слабое звено? Легко: тяните цепь за оба конца, пока она не разорвется. Где она разорвется? На самом слабом звене. Если в цепи сто звеньев, а современное воздушное судно может содержать более ста авионических систем, то цепь прочна ровно настолько, насколько прочно ее самое слабое звено. С точки зрения безопасности нет пользы в том, что девяносто девять звеньев чрезмерно прочны, если одно звено слабее, чем необходимо.

Безопасность полетов связана с определением “прочности”, или надежности, каждого аспекта авиационной экосистемы при понимании того, что общая безопасность зависит от множества разных аспектов и их взаимосвязей. Ее нельзя оценить, просто нагружая систему до катастрофы: задача безопасности состоит в предотвращении катастроф, а не в их создании. Поэтому безопасность полетов должна опираться на научный подход, использующий анализ, инженерные методы, процессы и стандарты. Воздушное пространство, связь, воздушные суда, системы и компоненты важны все, но каждое из этих направлений дает свою точку зрения, которую нужно учитывать при оценке безопасности. С каждой такой точкой зрения связаны многочисленные критерии, влияющие на безопасность; каждый критерий важен, хотя вклад отдельных критериев может различаться. (Отсюда следует обсуждение уровней гарантии разработки (Development Assurance Levels, DAL), обычно называемых уровнями критичности; наиболее подробно они рассматриваются далее в разделе о безопасности по ARP4761/A, где ARP означает аэрокосмическую рекомендуемую практику (Aerospace Recommended Practice).) Общая структура авиационной сертификации показана на следующем рисунке:



Экосистема авиационной сертификации

Как показано на рисунке выше, авиационная сертификация начинается в левом верхнем углу с процесса оценки безопасности на основе ARP4761/A. Архитектуры воздушного судна и авионических систем определяются вместе с требованиями безопасности, которые затем передаются в процесс разработки и сертификации воздушного судна и систем на основе ARP4754A. Далее нижележащие аспекты авионических систем рассматриваются с помощью различных руководств серии DO-XXX, где DO здесь означает “североамериканский документ” (North American Document). В Европе существует похожая экосистема, но такие документы обозначаются как ED-XXX, где ED означает “европейский документ” (European Document). Остальная часть книги посвящена техническому введению в каждый из этих документов и его разбору, чтобы объяснить роль документа в экосистеме авиационной разработки и сертификации.

Проектирование и сертификация авиационных систем отчасти похожи на проектирование и изготовление звеньев цепи: изготовитель должен знать назначение цепи и требуемую прочность. Без этого знания он сделает звено либо слишком прочным, напрасно потратив ресурсы, либо слишком слабым, из-за чего звено разрушится преждевременно. Оба исхода считаются неудачей. Неспециалисту может казаться, что в авионике нормально просто делать изделие как можно ближе к совершенству и с максимально возможной надежностью. Но в сложных системах “совершенства” не существует. Без заданного минимального уровня эксплуатационной надежности и целостности невозможно определить, достаточно ли хороша система. Надежность без защитных мер не полностью решает вопросы гарантии качества изделия, характера

отказов и их влияния на уровне системы. Если цепь разрывается, как это влияет на эксплуатационную систему? Надежность сама по себе задает лишь частоту отказов; важно еще и то, как именно система отказывает – управляемо или нет. Звено цепи может разрушиться, но проблему безопасности создают способ разрушения и последствия этого отказа. Поэтому нужна еще и составляющая целостности: нельзя просто “полагаться” на надежность. Любое спроектированное и изготовленное изделие в какой-то степени несовершенно. Именно такие ситуации, когда сочетание несовершенств складывается по закону Мерфи, и должна выявлять оценка безопасности. Поэтому разработка авионики, по сути, становится изучением экосистемы разработки авионики, то есть предметом этой книги.

На действительно свободном рынке, без основанных на безопасности регуляторных требований, стоимость и качество постоянно снижались бы до тех пор, пока не разорвалось бы звено в цепи безопасности полетов, с катастрофическими последствиями. Вместо этого авионические системы анализируют, чтобы понять влияние отказов на безопасность воздушного судна и людей. Затем определяют связь между корректной работой этой системы и общей безопасностью и назначают для системы минимальный порог надежности. После этого применяются инженерные методы, обеспечивающие необходимую строгость исходя из заданного минимального уровня надежности и целостности.

Замечание о стоимости авиационной разработки и сертификации по сравнению с безопасностью.

Разработка авионики недешева: почти всегда разработка изделий с доказуемыми характеристиками качества обходится дороже, чем разработка изделий без таких характеристик. Во всем мире распространено ошибочное представление, будто авионика дорога из-за чрезмерно обременительного процесса разработки и сертификационных требований. Нет сомнений, что строгие требования к разработке и сертификации увеличивают первоначальные затраты; иначе и быть не может. Однако это расхожее утверждение упускает два ключевых вопроса, связанных со стоимостью разработки авионики:

1. Учитывая важность безопасности в авиации, являются ли чрезмерными дополнительные затраты, связанные с разработкой и сертификацией авионики?
2. Действительно ли в отрасли возможен эффект масштаба, если успешные проекты дают сотни проданных единиц, тогда как в потребительской электронике успех измеряется миллионами единиц?

Следовательно, эти два вопроса сильно влияют на область анализа “затраты-выгоды”, а возможно, и полностью меняют правила игры. Как и в политике или в споре о пицце, по этим вопросам никогда не будет полного согласия. Но в одном согласны все: разработка авионики не может быть дешевой, зато она может быть экономически эффективной. Кроме того, авиация, пожалуй, сильнее любого другого рынка делает акцент на качестве, а

не на цене: дешевое изделие с отказами, большим объемом технического обслуживания или частой заменой не является ни дешевым, ни экономически эффективным. Покупали ли вы когда-нибудь дешевый инструмент, чтобы затем разочароваться в его работе и позже заменить его более дорогим аналогом? Для авионики эта ситуация весьма показательна: затраты на разработку изделия могут исчисляться миллионами долларов, а себестоимость единицы продукции – десятками тысяч.

Сами воздушные суда являются воплощением компромисса: стоимость, расход топлива и полезная нагрузка неразрывно связаны. Поэтому сегодня продаются сотни разных новых моделей воздушных судов: каждый тип хорошо подходит для определенной потребности, но требует компромиссов по сравнению с другими типами. Увеличение полезной нагрузки повышает цену покупки и эксплуатационные расходы. Уменьшение полезной нагрузки экономит топливо, но позволяет перевозить меньше груза или пассажиров, что отрицательно влияет на выручку. В каждой категории обычно есть несколько конкурентов; отсюда и сотни разных моделей воздушных судов на рынке.

С безопасностью авионики действует похожая зависимость. Дополнительную надежность, или безопасность, можно обеспечить, но всегда ценой дополнительных затрат. Например, если разрабатывать резервные бортовые системы спутниковой связи так, будто они являются реверсорами тяги, стоимость существенно вырастет, а общая безопасность увеличится незначительно, если вообще увеличится. И снова все сводится к компромиссам. Поэтому безопасность в авиации сочетает искусство и науку, а научная часть включает анализ “затраты-выгоды” для каждого предлагаемого аспекта, связанного с безопасностью. При неограниченном времени и бюджете безопасность всегда можно улучшать дальше. Вспомните старые военные самолеты с восемью двигателями. Почему восемь? Потому что “конечно же” восемь должны быть надежнее шести, шесть надежнее четырех и так далее. А сколько двигателей у самого современного серийного истребителя в мире? У истребителя F-35 Joint Strike Fighter один двигатель. Именно. Теперь рассмотрим пример электродистанционной системы управления, ЭДСУ (Fly-By-Wire, FBW): обычные тросовые и шкивные системы заменили ЭДСУ, но какой ценой? Одноканальные (simplex, single-channel) электронные средства и исполнительные механизмы обычно не удовлетворяют требованиям безопасности, поэтому архитектуры приходится удваивать, затем, возможно, утраивать, а иногда и учетверять, чтобы выполнить эти требования. Это геометрически увеличивает стоимость, массу, сложность и, разумеется, затраты. Поэтому во всей экосистеме разработки авионики вопросам стоимости уделяется прагматическое внимание.

Последнее слово перед погружением в материал...

В конечном счете проектов с неограниченным временем и бюджетом не бывает. Авионика называется авионикой потому, что должна поддерживать полет и соответствовать системе обеспечения безопасности, объясняемой в этой книге. Далее изложен материал, который помогает успешно пройти путь компромисса “затраты-выгоды” и при этом достичь достаточного уровня безопасности.

Сертификаты типа (ТС), технические стандартные приказы (TSO) и одобрение изготовителя частей (PMA)

Автор: Вэнс Хилдерман

Рецензенты:

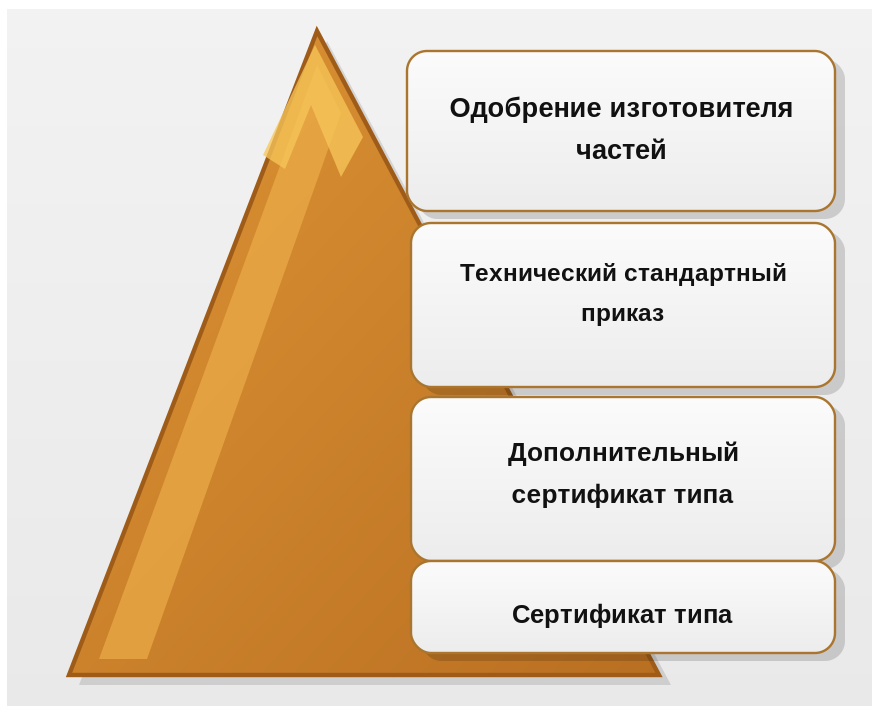
- д-р J. Lee, менеджер по авионике, полковник ВВС Республики Корея в отставке
- Antonio Silva (магистр наук по электротехнике (Master of Science in Electrical Engineering, MSEE), профессиональный инженер (Professional Engineer, PE)), руководитель направления разработки ПО (Software, SW) в AgustaWestland, инженер по верификации соответствия по ПО и бортовой электронной аппаратуре (Airborne Electronic Hardware, АЕН)

Итак, вы понимаете, что бортовая электронная аппаратура входит в безопасную систему и что в инженерном жизненном цикле есть обязательные руководящие материалы. Значит, можно начинать проектировать авионическую систему? Не так быстро. Сначала нужны полные ответы на следующие вопросы. Подсказка: эти ответы лежат в понимании технических стандартных приказов (Technical Standard Order, TSO), сертификатов типа (Type Certificate, TC), дополнительных сертификатов типа (Supplemental Type Certificate, STC) и одобрений изготовителя частей (Parts Manufacturer Approval, PMA).

- a. Можно ли обычно начинать разработку авионической системы, не зная, на какой тип воздушного судна она будет установлена?
- b. Существуют ли специальные стандарты, зависящие от функций авионической системы?
- c. Если авионическая система уже была сертифицирована для одного воздушного судна, всегда ли нужно сертифицировать ее заново для другого типа воздушного судна?
- d. Можно ли изготавливать блоки авионики на любом производстве высокого качества?

Если на эти четыре вопроса вы ответили “нет”, “да”, “нет” и “нет”, значит, у вас уже есть хотя бы базовое представление о технических стандартных приказах, сертификатах типа, дополнительных сертификатах типа и одобрениях изготовителя частей. В любом случае контекст ответов на эти базовые вопросы важен для понимания экосистемы разработки авионики.

Для коммерческой авионики безопасность полетов имеет первостепенное значение. Пожалуй, нигде в экосистеме авионики взаимозависимость этих аспектов безопасности не видна так наглядно, как в пирамиде летной годности с четырьмя строительными блоками, показанной ниже:



Пирамида летной годности

Как и у обычной пирамиды, у пирамиды летной годности есть основание, а остальные блоки располагаются друг над другом. Сначала разберемся, что такое летная годность.

Одобрение изготовителя частей: объединенное одобрение конструкции и производства для модификационных и заменяемых частей. Оно позволяет изготовителю производить и продавать такие части для установки на изделия, имеющие сертификат типа.

Одобрение изготовителя частей (PMA)

Летная годность – это доказуемая способность воздушного судна безопасно эксплуатироваться при выполнении заранее определенных функций. Когда летная годность коммерческого воздушного судна доказана, может быть выдан сертификат летной годности. Федеральное управление гражданской авиации США (Federal Aviation Administration, FAA) определяет его как документ FAA, разрешающий эксплуатацию воздушного судна в полете (www.faa.gov). Согласно приказу FAA 8130, для получения сертификата летной годности должны выполняться оба условия ниже:

- a. *Воздушное судно должно соответствовать своей типовой конструкции. Считается, что соответствие типовой конструкции достигнуто, когда конфигурация воздушного судна, двигателя, воздушного винта и установленных изделий согласуется*

с чертежами, спецификациями и другими данными, входящими в состав сертификата типа. Это включает любой дополнительный сертификат типа, а также ремонты и доработки, внесенные в воздушное судно.

- в. Воздушное судно должно находиться в состоянии, безопасном для эксплуатации. Это относится к состоянию воздушного судна с точки зрения износа и ухудшения состояния, например коррозии обшивки, расслоения/помутнения остекления, утечек жидкостей и износа шин, а также любых невыполненных директив летной годности.

Таким образом, пирамида летной годности состоит из четырех строительных блоков. Ниже они кратко перечислены, а затем рассматриваются подробнее:

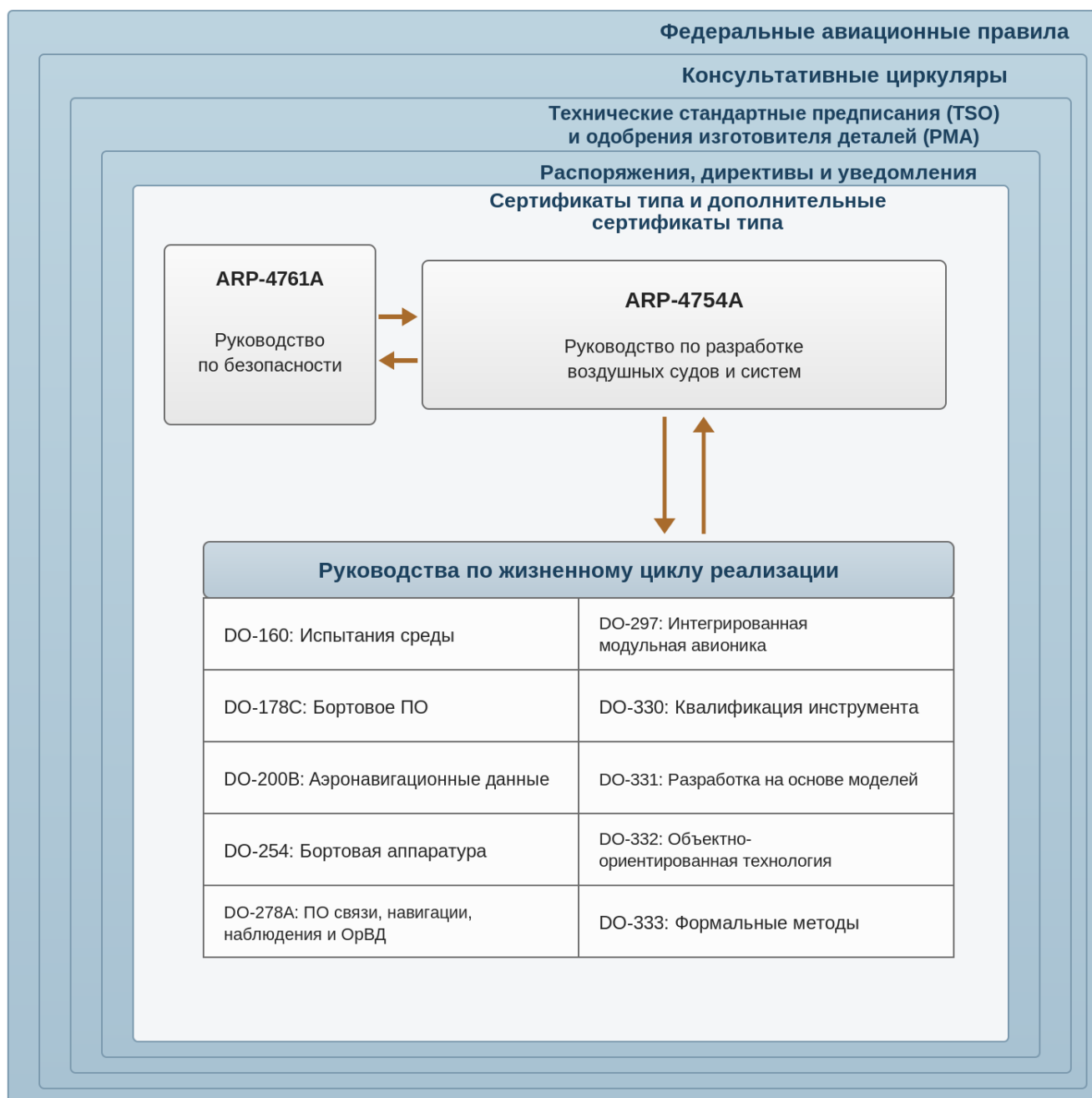
Пирамида летной годности Итак, у пирамиды летной годности есть четыре важных взаимосвязанных строительных блока, которые подробнее рассматриваются ниже.



Пирамида летной годности

Помните: Экосистема авиационной разработки и сертификации

Помните: при разработке авионики все сводится к экосистеме. Ниже снова показаны ее ключевые аспекты. Какой из них важнее остальных? Правильный, хотя и не самый удобный ответ: важны все. Как цепь прочна настолько, насколько прочно ее самое слабое звено, так и система ограничена самым слабым элементом своей экосистемы. На следующем рисунке показана экосистема авиационной разработки и сертификации:



Подробности: сертификаты типа, дополнительные сертификаты типа, технические стандартные приказы и одобрения изготовителя частей...

Прежде всего, какова формальная нормативная основа сертификации воздушных судов и авионики? Основные положения содержатся в разделе 14 Свода федеральных нормативных актов США (Code of Federal Regulations, CFR) “Аэронавтика и космос”, а именно в частях:

- 21 (Процедуры сертификации изделий и частей)
- 23 (Нормы летной годности: самолеты нормальной, многоцелевой, акробатической и пригородной категорий)
- 25 (Нормы летной годности: самолеты транспортной категории)
- 27 (Нормы летной годности: винтокрылые летательные аппараты нормальной категории)
- 29 (Нормы летной годности: винтокрылые летательные аппараты транспортной категории) и соответствующие уровни поправок, которые обычно определяют основной базис сертификации

Сертификат типа (ТС)

Сертификат типа образует основу летной годности, поскольку подтверждает состоятельность конкретной заданной производственной конструкции. Если вспомнить доцифровую эпоху, сертификат типа действительно был именно сертификатом: документом, который выдает уполномоченный орган, например FAA, Агентство Европейского союза по авиационной безопасности (EASA), Управление гражданской авиации (Civil Aviation Authority, CAA) и т. д. Но сертификат типа – не просто лист бумаги. Он задает применимые критерии сертификации, которые зависят от типа одобряемого изделия и от действующих нормативных требований, включая применимые поправки. Сертификат типа включает достаточные сведения или ссылки на них, чтобы показать: одобряемая конструкция соответствует применимым требованиям летной годности. Что такое применимые требования? Для каждого типа воздушного судна или оборудования действует свой набор требований. Он различается по типам, но обычно включает технические стандартные приказы, а также соответствие DO-178C (ПО), DO-254 (бортовая электронная аппаратура), DO-160 (условия внешней среды и электромагнитные помехи (Electromagnetic Interference, EMI)) и т. д. Сертификат типа включает таблицу данных сертификата типа (Type Certificate Data Sheet, TCDS), которая идентифицирует типовую конструкцию, эксплуатационные ограничения, держателя сертификата типа, изготовителя и модель, а также определяет применимые правила или ограничения, установленные сертифицирующим органом, включая уровни поправок к правилам. Из компонентов право на сертификат типа имеют только двигатели и воздушные винты. Сертификат типа служит основой для последующих одобрений, например производственного одобрения и подтверждения летной годности.

На самом деле есть две ключевые причины, по которым сертификаты типа образуют основу летной годности:

1. Сертификаты типа определяют применимые правила, относящиеся к каждому изделию, и фиксируют, что соответствие этим правилам было подтверждено.
2. Сертификаты типа требуются до фактического начала производства такого изделия.

План сертификации для сертификата типа (ТС)

Как и при строительстве дома, трудно доказать соответствие строительным нормам, если инспекторы не участвовали до начала строительства и во время него. Поэтому для большинства зданий сначала рассматривают план, а затем проводят поэтапные проверки по мере готовности фундамента, стен, электрики и крыши. То же относится к воздушным судам и авионике, включенным в сертификат типа. Заявитель на сертификат типа должен представить сертифицирующему органу подробный план сертификации и привлекать этот орган или его представителя на протяжении всего проекта. Что входит в план сертификации? Достаточно сведений, чтобы показать: заявитель глубоко понимает применимые нормы и способен реализовать безопасную конструкцию так, чтобы ее можно было проверить. В частности, план сертификации для сертификата типа должен подробно описывать приведенные ниже сведения и быть одобрен сертифицирующим органом до перехода к инженерной реализации, то есть к фактической разработке:

- Конкретная информация о том, кто подает заявку, дата и идентификация конфигурации.
- Описание конструкции или изменения конструкции, особенно для дополнительного сертификата типа (см. ниже), с достаточной детализацией схем и диаграмм, чтобы показать инженерную безопасность.
- Эксплуатационная среда, допущения, ограничения и пределы.
- Базис сертификации, включая применимые нормы (ARP-47XX, DO-XXX и т. д.), например части 23, 25, 27 или 29 правил FAA (14 CFR Parts 23, 25, 27 или 29; в зависимости от воздушного судна могут применяться и другие части).
- Описание средств подтверждения соответствия, их пригодности и того, как артефакты с управляемой конфигурацией будут использоваться для такого подтверждения.
- Перечень представляемых документов, включая все артефакты, требуемые применимыми нормами.
- Перечень испытательных изделий, используемых для получения данных о соответствии, включая любые специальные указания или характеристики.
- Описание того, как после выдачи сертификата типа будут выполняться требования к поддержанию безопасности эксплуатации.

- Предлагаемый график проекта с указанием ключевых вех, включая все мероприятия, требующие надзора или присутствия сертифицирующего органа.
- Если участвует FAA, указание всех представителей FAA по инженерной экспертизе (Designated Engineering Representatives, DER), назначенных представителей по летной годности (Designated Airworthiness Representatives, DAR), назначенных представителей по производственной инспекции (Designated Manufacturing Inspection Representatives, DMIR) и организационных назначенных представителей по летной годности (Organizational Designated Airworthiness Representatives, ODAR), которых планируется задействовать в проекте сертификации, вместе с областями ответственности и полномочиями.

Заявитель на сертификат типа представляет документы в местный авиационный регулирующий орган, а тот в итерационном порядке дает замечания, пока планы не будут одобрены. После этого заявитель работает по этим планам, а сертифицирующий орган или его представители регулярно проверяют ход работ. Как и при строительстве дома, проверки идут поэтапно, чтобы рано видеть фактический прогресс относительно плана и анализировать отклонения. На практике представление планов обычно связано с формальной заявкой изготовителя оригинального оборудования (Original Equipment Manufacturer, OEM) на сертификат типа. Это происходит, когда воздушное судно считается достаточно зрелым, что в том числе подтверждается экспериментальными полетами по процедурам выдачи разрешения на полет, и есть уверенность, что сертификация будет достигнута к определенной дате. Так выглядит типичный процесс для сертификата типа. Для технического стандартного приказа или его европейского аналога процесс другой: документы представляют в офис сертификации воздушных судов (Aircraft Certification Office, ACO) или в EASA, а рассмотрение идет перед регулирующим органом и не зависит от сертификации конкретного воздушного судна. Сертифицированное воздушное судно может использоваться как летающая испытательная лаборатория для конкретной авионики, представляемой на рассмотрение.

В Европе представление выполняется непосредственно в профильную центральную дирекцию EASA. Затем агентство определяет организацию управления программой сертификации (Program Certification Management) и профильные комиссии, по одной на каждую специальность. В них могут входить специалисты центральных подразделений EASA и/или специалисты национального авиационного органа заявителя. Если заявитель получил одобрение проектной организации (Design Organization Approval, DOA) и назначил инженеров по верификации соответствия (Compliance Verification Engineers, CVE) по различным специальностям, EASA также определяет свой уровень участия (Level of Involvement, LOI) в аудите каждого проверяемого подкомпонента. Сам аудит в любом случае выполняют инженеры по верификации соответствия, а специалисты EASA участвуют в соответствии с установленным уровнем участия. Сертификат типа воздушного судна обычно валидируется национальными органами по летной годности,

когда воздушное судно продается заказчику, который намерен эксплуатировать его за пределами США или Европейского союза. Валидация между США и ЕС упрощается двусторонними соглашениями между FAA и EASA.

Авиационные нормы постоянно пересматриваются. Что тогда происходит с проектами сертификации при изменении норм? Чтобы избежать почти бесконечного цикла обновления проекта под меняющиеся требования, нормативный базис считается замороженным на разумный период реализации проекта. Поэтому нормативные требования, указанные в одобренных планах, становятся базисом сертификата типа, даже если эти нормы изменятся до завершения окончательной сертификации. Из этого правила есть исключения, но обычно оно предотвращает ситуацию, когда заявителю пришлось бы менять конструкцию только из-за непредвиденного изменения норм.

Изменения конструкции

Существуют только два типа конструкций авионики: те, которые уже изменились, и те, которые еще изменятся. (Вэнс Хилдерман, 2014)

Коммерческой авиации едва исполнилось сто лет, и ее развитие продолжается непрерывно. Развитие означает изменения. Изменения означают ту или иную форму повторной сертификации. Что происходит, когда нужно изменить уже одобренную конструкцию по сертификату типа? Обычно есть три варианта:

Простая повторная сертификация: изменение считается простым и не требует обширного анализа влияния на безопасность. По сути, это модификация сертификата типа. Классификация изменения зависит только от его возможного влияния на безопасность. Пример: добавление коротковолновой радиостанции при сохранении двух стандартных ультракоротковолновых радиостанций активными и установленными. В таком случае нужно проверить только правила установки, электромагнитную совместимость (ЭМС; Electromagnetic Compatibility, EMC) и электромагнитные помехи: дополнительная коротковолновая радиостанция считается оборудованием без опасности, пока ее работа не влияет на другое оборудование в конфигурации сертификата типа. Обратите внимание: повторная сертификация для незначительных и менее значимых изменений может выполняться держателем технического стандартного приказа или держателем сертификата типа, имеющим одобрение проектной организации. Во всех остальных случаях требуется повторная сертификация с участием сертифицирующего органа.

Повторная сертификация через дополнительный сертификат типа: изменение может быть непростым, но не вводит новые конструкции с самостоятельными аспектами безопасности. Сегодня дополнительный сертификат типа понимается как добавление к существующему сертификату типа, то есть как построение поверх одобренной конструкции. Обычно речь идет о добавлении функциональности авионики или

установочных комплектов в конфигурацию воздушного судна. Дополнительный сертификат типа применяется к конкретно указанным серийным номерам воздушных судов.

Создание полностью новой конструкции: повторение процесса сертификации с использованием многих уже существующих артефактов, то есть ранее разработанных данных типовой конструкции. Это применимо к авионике, но не к воздушному судну в целом: для данного воздушного судна существует только один сертификат типа. Если оформляется другой сертификат типа, значит, по определению это уже другой тип воздушного судна.

Казалось бы, безопасность в авиации была бы максимальной, если бы каждый раз начинать заново и создавать новую конструкцию, верно? Совсем нет. Старт с чистого листа просто дает новым ошибкам путь в полностью новую конструкцию и одновременно делает ее настолько дорогой, что это отбивает желание внедрять новые возможности, связанные с безопасностью. Если изменение особенно простое, самым простым выбором будет вариант 1 выше. Если изменение может ввести новые риски безопасности, не рассмотренные в ранее одобренной конструкции, сертифицирующий орган может потребовать более тяжелый вариант 3. Но если изменение можно удержать в пределах ранее одобренного контекста безопасности, тогда можно использовать дополнительный сертификат типа, как описано ниже.

Сохранение действительности сертификата типа (ТС)

Держатель сертификата типа должен поддерживать его действительность: постоянно выполнять директивы летной годности (Airworthiness Directives, AD), управлять сервисными бюллетенями и при необходимости предоставлять техническую поддержку, почти всегда за плату. Обычно это называется поддержкой типа и часто продолжается после прекращения производства изделия. Те же правила применяются и к дополнительным сертификатам типа, описанным ниже. Если держатель сертификата типа или дополнительного сертификата типа прекращает поддержку изделия, сертификат возвращается сертифицирующему органу, а использование этого изделия в коммерческой эксплуатации воздушных судов после этого запрещается.

Дополнительный сертификат типа (STC)

Дополнительный сертификат типа – это сертификат типа для изделия, которое уже имеет ранее одобренный сертификат типа, но было модифицировано. Когда изготовитель показывает, что модификация не вводит риски, не рассмотренные в исходной конструкции, он обычно запрашивает дополнительный сертификат типа, а не совершенно новый сертификат типа, потому что дополнительный сертификат типа требует меньше работы. Такой сертификат подробно описывает характер изменения конструкции воздушного судна и то, как модификация влияет на ранее одобренную конструкцию, включая применение существующих норм. Объем дополнительного сертификата типа

может сильно различаться в зависимости от характера модификации. Это типичный инструмент, с помощью которого новые авионические системы, то есть авиационные изделия, устанавливаются на воздушное судно с действующим сертификатом типа. Такая установка требует проектирования и изменения воздушного судна, чтобы соответствующее изделие можно было установить на него. Как и в процессе сертификата типа, разрабатывается план сертификации конкретного проекта (Project Specific Certification Plan, PSCP), который определяет мероприятия по дополнительному сертификату типа и базис сертификации. Обычно такие проекты идут по установленному процессу: планирование, проектирование установки, установка, предварительные сертификационные испытания, получение разрешения на типовую инспекцию (Type Inspection Authorization, TIA), включая оценку риска летных испытаний, подтверждение соответствия установки, официальные наземные испытания для сертификационного зачета и официальные летные испытания для сертификационного зачета. В рамках этого процесса до официальной типовой инспекции, включая подтверждение соответствия, наземные и летные испытания, может быть выдан экспериментальный сертификат летной годности (Experimental Airworthiness Certificate, EAC) или разрешение на полет (Permit To Fly, PTF), как это называется в Европе. Соответствие нормативным требованиям обосновывается разработкой набора документов; ниже приведен типичный перечень документов для проекта дополнительного сертификата типа:

1. Заявка на дополнительный сертификат типа
2. План сертификации конкретного проекта
3. План подтверждения соответствия
4. Главный перечень данных (Master Data List, MDL) (форма DER 8110-3, одобряющая этот документ)
5. План сертификации по аспектам человеческого фактора (Human Factors Certification Plan, HFCP), при необходимости
6. Инструкции по поддержанию летной годности (Instructions for Continued Airworthiness, ICA)
7. Дополнение к руководству по летной эксплуатации самолета (Aircraft Flight Manual Supplement, AFMS) (форма DER 8110-3, рекомендуемая одобрение этого документа)
8. Контрольный перечень матрицы соответствия сертификационным требованиям (рассмотрение DER)
9. Процедура и план наземных испытаний на электромагнитные помехи и радиочастотные помехи (Radio Frequency Interference, RFI) (форма DER 8110-3, одобряющая этот документ)
10. Процедура и план функциональных наземных испытаний (форма DER 8110-3, одобряющая этот документ)

11. Процедуры и план летных испытаний (форма DER 8110-3, рекомендуемая одобрение этого документа)
12. Запрос на разрешение на типовую инспекцию
13. Электрическая схема
14. Инструкции по установке
15. Трассировка проводов
16. Чертеж жгута проводов
17. Схема установки оборудования
18. Анализ электрической нагрузки
19. Специальные условия / документ по проблемному вопросу (Special Conditions / Issue Paper) (требуется только для уникальных, новых или таких дополнительных сертификатов типа, где нужно подтвердить соответствие в областях, пока не регулируемых нормами FAA)
20. Перечень установочного комплекта
21. Чертежи табличек и надписей
22. Отчет по структурному обоснованию
23. Масса и центровка
24. Дополнительный отчет по летным испытаниям (после типовой инспекции) (форма DER 8110-3, одобряющая этот документ)
25. Дополнительная инспекция соответствия воздушного судна
26. Анализ безопасности системы (System Safety Analysis, SSA) на уровне воздушного судна (форма DER 8110-3, одобряющая этот документ)
27. Механические / структурные чертежи (по применимости)
28. Чертежи приемников полного и статического давления, по применимости
29. Письмо-разрешение владельца воздушного судна на экспериментальный сертификат летной годности
30. Программное письмо по экспериментальному сертификату летной годности
31. Заявка на экспериментальный сертификат летной годности (форма 8130-6)
32. Подтверждение соответствия установки и заявление о соответствии (8130-9)

Обычно большинство, а часто и все перечисленные выше элементы данных должен разработать заявитель на дополнительный сертификат типа, чтобы обосновать соответствие нормативным требованиям. Таким заявителем может быть владелец одного или нескольких воздушных судов либо отдельная организация, обычно крупный поставщик авионики, не обязательно совпадающий с исходным держателем сертификата типа. В некоторых случаях для воздушного судна по части 23 правил FAA (14 CFR Part 23) строгость и объем работ по сертификационным требованиям дополнительного сертификата типа ниже, чем для воздушного судна по части 25 правил FAA (14 CFR Part 25). После завершения всего пакета подтверждения соответствия и его представления регулирующему органу можно завершить окончательную обработку, после чего заявителю выдается дополнительный сертификат типа.

Технический стандартный приказ (TSO)

Технический стандартный приказ или европейский технический стандартный приказ (European Technical Standard Order, ETSO) представляет собой уникальный набор стандартов характеристик и интерфейсов, то есть минимальных стандартов характеристик (Minimum Performance Standards, MPS), для определенного изделия, используемого на коммерческих воздушных судах. Одобрение по техническому стандартному приказу (TSO Authorization) означает, что сертифицирующий орган рассмотрел изделие и подтвердил: оно соответствует минимальным стандартам характеристик этого приказа, то есть требованиям к конструкции и производству, а изготовитель может переходить к производству. Поэтому такой приказ включает как одобрение конструкции, так и одобрение производства. Однако в зависимости от типа изделия технический стандартный приказ дает только одобрение конструкции и производства; он не дает одобрения установки. Одобрение установки нужно получать отдельно, и само по себе оно не обязательно означает возможность перейти к установке на воздушное судно.

Обычно на борту воздушного судна находятся десятки различных компьютерных систем, а на крупных коммерческих воздушных судах – более ста уникальных систем. Поскольку каждая система выполняет свои функции, для многих таких систем нужно показать, что они обеспечивают базовые возможности, общие для данного типа системы. Поэтому эти базовые возможности называют минимальными стандартами характеристик. Технические стандартные приказы закрепляют такие минимальные стандарты характеристик и интерфейсов для соответствующего типа системы.

Технические стандартные приказы охватывают многие изделия, типичные для гражданских воздушных судов, включая авионические системы. В авионике они распространяются на основные пилотажные приборы и другие системы: автопилоты, радиостанции, предупреждение о сваливании, вычислители воздушных параметров, системы пространственного положения и курса (Attitude Heading Reference System, AHRS), системы воздушных данных, пространственного положения и курса (Air Data and Attitude Heading Reference System, ADAHRS), системы предупреждения о сдвиге ветра, системы управления полетом (Flight Management System, FMS), приемники глобальной

системы позиционирования (Global Positioning System, GPS) и т. д. Некоторые устройства соответствуют только части требований технического стандартного приказа; такие случаи называют частичными техническими стандартными приказами. Как следует из названия, они относятся к устройствам, обеспечивающим часть полной функции такого приказа. Заявители, разрабатывающие устройство с частичным техническим стандартным приказом, должны заранее, до начала каких-либо работ по разработке, согласовать свои планы с сертифицирующим органом, чтобы тот принял базис сертификации и базис такого частичного приказа.

Технические стандартные приказы выпускаются регулирующим органом и обычно опираются на документы серии DO, разрабатываемые федеральными консультативными комитетами, такими как Радиотехническая комиссия по авиации (Radio Technical Commission for Aeronautics, RTCA). В разработку документов серии DO вносит вклад слабосвязанная группа международных организаций, пользователей и регулирующих органов. Эти документы, например RTCA/DO-178C “Соображения по ПО при сертификации бортовых систем и оборудования” (Software Considerations in Airborne Systems and Equipment Certification), становятся неотъемлемой частью требований технического стандартного приказа и обычно упоминаются в нем.

Одно важное замечание: технические стандартные приказы являются важным официальным признанием одобрения со стороны FAA. Поэтому они выдаются иностранным изготовителям только при выполнении всех требований такого приказа и при производстве в стране, с которой заключено признанное двустороннее соглашение, применимое к рассматриваемому изделию авионики. Например, на момент написания этой книги у США нет общих двусторонних соглашений с Турцией, тогда как у EASA они есть. Поэтому турецкие изготовители часто могут подать заявку на европейский технический стандартный приказ и получить его, тогда как получить американский технический стандартный приказ сложнее.

Значение технического стандартного приказа (TSO)

Помимо того что технические стандартные приказы являются обязательным стандартом для распространенных изделий авионики, они важны еще по нескольким причинам:

Важные аспекты технических стандартных приказов и специальные европейские соображения

- Стандартизировать общие функции, входы и выходы.
- Повысить согласованность между производителями и воздушными судами.
- Повысить безопасность и надежность.
- Задать минимальные характеристики по типам систем.
- Упростить обновление технических стандартов Федерального управления гражданской авиации США (Federal Aviation Administration, FAA) при типовых улучшениях конструкции.

Таким образом, технические стандартные приказы являются важным слоем пирамиды летной годности для авионики: они задают общий набор дополнительных эксплуатационных характеристик для распространенных авионических систем. Для более сложных или менее зрелых авионических систем такие приказы обычно обновляются чаще, постепенно развивая минимальные эксплуатационные требования для данного типа системы. Обновление технического стандартного приказа может отменять прежние версии, а может и не отменять их; обновленный документ прямо укажет, как именно обстоит дело. Следует отметить, что EASA теперь считает одобрения американских технических стандартных приказов имеющими ограниченную ценность не только тогда, когда такой приказ устарел, но и в целом. Причина в том, что процесс АСО по техническим стандартным приказам, особенно для изменений, иногда воспринимается как быстрый способ получить одобрение. Особенно это заметно, когда оборудование обеспечивает 10 % функциональности по минимальным эксплуатационным стандартам характеристик (Minimum Operational Performance Standards, MOPS) технического стандартного приказа, а остальные 90 % составляют дополнительные функции. EASA требует для каждого оборудования с техническим стандартным приказом, даже с европейским техническим стандартным приказом, заполнять опросники о соблюдавшихся нормах, включая консультативные циркуляры и документы серии DO. Отдельно требуется объяснить, как была обеспечена гарантия разработки для функций, не определенных техническим стандартным приказом. Действуйте со знанием дела и осторожностью.

Пример технического стандартного приказа (TSO)

Чтобы понять основы технического стандартного приказа, рассмотрим TSO-C63d под названием “Бортовое метеолокационное оборудование” (Airborne Weather Radar Equipment). В названии есть аббревиатура TSO, значит, это явно технический стандартный приказ (Technical Standard Order). Классификатор C63 – уникальное обозначение, присвоенное бортовому метеолокационному оборудованию. Буква редакции d означает пятую редакцию TSO-C63, поскольку вторая редакция была бы обозначена буквой a в C63a. Документ TSO-C63d имеет объем 15 страниц и указанную дату вступления в силу (Effective Date) 2/28/12. Это означает, что новые сертификации

метеолокаторов после 28 февраля 2012 года должны соответствовать стандартам, указанным в TSO-C63d. Что определяет этот документ? Его ключевое содержание включает:

- a. **Назначение (Purpose).** Описывает, какие функции охватывает этот технический стандартный приказ; в данном случае это возможность обнаружения сдвига ветра впереди по курсу. Функции системы управления полетом специально исключены, поскольку рассматривается в другом месте.
- b. **Область применения (Applicability).** Указывает, что этот технический стандартный приказ предназначен для новых бортовых метеосистем, разработка которых планируется после даты вступления в силу (28 февраля 2012 года), и что предыдущие версии этого приказа отменяются.
- c. **Требования (Requirements).** Подробное перечисление требований: минимальные функции бортового метеоборудования, классификации отказных состояний, функциональная и экологическая квалификация по RTCA DO-173 и DO-220, квалификация ПО и аппаратуры по RTCA DO-178 и DO-254, требования к данным и документации.
- d. **Минимальные стандарты характеристик (Minimum Performance Standards).** Фактические критерии измерения и представления применимых метеоданных для обнаружения сдвига ветра, а также подробности моделирования и испытаний этих функций.

Документ TSO-C63d является типичным техническим стандартным приказом по объему, формату и организации. Но, как и все такие приказы, его содержание специально адаптировано к потребностям рассматриваемой системы, то есть метеолокатора (Weather Radar). В данном случае речь идет о функции обнаружения сдвига ветра в бортовых метеолокационных системах.

Одобрение изготовителя частей (PMA)

Одобрение изготовителя частей – это объединенное одобрение конструкции и производства для модификационных и заменяемых частей. Оно позволяет изготовителю производить и продавать такие части для установки на изделия, имеющие сертификат типа. Такое одобрение представляет собой одобрение системы изготовления и контроля (Fabrication and Inspection System, FIS) изготовителя. Устройство, одобренное по техническому стандартному приказу, не требует от заявителя отдельного одобрения изготовителя частей, поскольку технический стандартный приказ уже является одобрением конструкции и производства. Одобрение изготовителя частей можно получить для заменяемой части, используемой устройством с техническим стандартным приказом, но в таких случаях установщик должен разместить на устройстве собственную табличку модификатора. Части, произведенные по разовому дополнительному

сертификату типа или по процедуре полевого одобрения (Field Approval), не имеют права на такое одобрение. Стандартные части, например гайки и болты, такого одобрения не требуют.

Для сертифицированных воздушных судов одобрение изготовителя частей обычно является необходимым основанием для установки устройств, которые могут влиять на безопасность. Для экспериментальных воздушных судов это правило не действует. Такое одобрение не является общим одобрением процедур инспекции, материалов или процессов; оно применимо только к конкретному устройству и конкретным процессам и процедурам для него. Изготовители, имеющие одобрение изготовителя частей, могут изготавливать и продавать заменяемые части, даже если они не были первоначальным изготовителем воздушного судна или изначально установленного изделия. Одобрение изготовителя частей может использоваться для производства модификационных частей по дополнительным сертификатам типа. В таких случаях после выдачи дополнительного сертификата типа его одобрение становится основанием для одобрения новой или модифицированной части, а одобрение изготовителя частей для этой новой или модифицированной части выдается вскоре после выдачи дополнительного сертификата типа.

Значение одобрения изготовителя частей росло по мере распространения и развития авиатранспорта. Владельцы и эксплуатанты воздушных судов не хотят зависеть от единственного поставщика в течение всего срока службы воздушного судна, поэтому им нужны альтернативные источники частей. С другой стороны, сертифицирующие органы понимают, что безопасность воздушного судна похожа на большую сложную цепь, где общий уровень безопасности определяется самым слабым звеном. Неодобренные части легко могут стать таким звеном и повлиять на безопасность.

Получение одобрения изготовителя частей обычно требует двух разных последовательных мероприятий, показанных ниже:

Два шага получения одобрения изготовителя частей (Parts Manufacturer Approval, PMA)

1. Доказать безопасность конструкции и соответствие применимым нормам и стандартам.
2. Получить одобрение производства, подтвердив надлежащую гарантию качества (Quality Assurance, QA).

Три метода получения одобрения конструкции части в рамках одобрения изготовителя частей

Метод 1. Одобрение изготовителя частей по идентичности через подтверждение лицензионного соглашения. Это не отдельный метод одобрения, а фактически способ обосновать идентичность:

Этот метод часто используется для получения одобрения изготовителя частей через лицензионное соглашение по дополнительному сертификату типа. В этом случае заявитель заключает лицензионное соглашение с держателем сертификата типа, дополнительного сертификата типа или технического стандартного приказа, а данные типовой конструкции из этих документов используются для обоснования идентичности.

Метод 2. Одобрение изготовителя частей (РМА) по идентичности без лицензионного соглашения:

В этом случае заявитель представляет заявление, удостоверяющее, что конструкция во всех отношениях идентична одобренной конструкции. Как правило, если у заявителя нет доступа к исходным данным конструкции, например для сложных или высокотехнологичных частей, получить одобрение изготовителя частей по идентичности без лицензионного соглашения почти невозможно.

Метод 3. Одобрение изготовителя частей (РМА) по испытаниям и расчетам:

Этот метод ровно таков, как звучит: это комплексный набор данных испытаний и расчетов, демонстрирующий соответствие применимым нормам летной годности.

Два последовательных мероприятия для получения одобрения изготовителя частей подробнее описаны ниже:

Шаг 1: доказательство безопасной конструкции. Сначала нужно показать, что изготавливаемая часть имеет безопасную конструкцию, то есть обладает летной годностью. Как показывается безопасная конструкция? Помните: сертификация авионики связана с доказательством своей невиновности, потому что, в отличие от обычных законов, безопасность полетов опирается на парадигму “виновен, пока не доказана невиновность”. Для одобрения изготовителя частей заявитель должен доказать безопасность конструкции следующим образом:

- a. Доказать, что для этой части были соблюдены все применимые сертификационные нормы и стандарты. В зависимости от части может применяться, например, сочетание DO-178 для ПО, DO-254 для аппаратуры, DO-160 для испытаний на воздействие внешней среды и т. д., а также консультативные циркуляры (Advisory Circular, AC) и, возможно, другие документы серии DO и требования.
- b. Если лицензионного соглашения по методу 1 выше нет, может использоваться дополнительный анализ, показывающий приемлемость части. Для простых конструкций такой анализ может состоять из инспекции части и последующего сравнения с аналогичными одобренными частями, чтобы показать, что новая часть имеет те же характеристики безопасности, что и уже одобренная. Для более сложных конструкций анализ может включать испытания и расчеты, напрямую доказывающие, что часть соответствует всем стандартам безопасности FAA. Обратите внимание: для

частей, содержащих аппаратную или программную логику, такой анализ не может заменить необходимость применения DO-254 или DO-178; этот вариант недействителен.

Шаг 2: получение производственного одобрения через надлежащую гарантию качества. После принятия шага 1 следующий шаг – получить фактическое производственное одобрение для части. Почему этот шаг нужен, если уже показано, что конструкция безопасна и соответствует всем стандартам и нормам? Ответ прост. Представьте человека, у которого есть “книга рецептов” здорового образа жизни: здоровая пища, упражнения, снижение стресса и сон. Само владение такой книгой едва ли гарантирует результат. Чтобы жить здорово, нужны дисциплина, время, деньги и отсутствие срывов на вредную пищу. То же верно и для безопасных авиационных частей: изготовитель должен доказать, что он следует безопасной конструкции и способен производить ее стабильно, имея меры по обнаружению и предотвращению дефектов. Такое доказательство вместе с обнаружением и предотвращением дефектов относится к функции гарантии качества (Quality Assurance, QA), с помощью которой доказывается соответствие. Это гарантирует, что части будут поставляться конечным пользователям только тогда, когда они соответствуют всем требованиям одобренной конструкции, норм и стандартов.

Одобрения изготовителя частей не передаются и остаются действительными, пока не будут добровольно возвращены, отозваны или прекращены, обычно сертифицирующим органом. Данные такого одобрения можно продать, обычно по лицензии, как упоминалось выше, но покупатель не получает автоматического разрешения на производство. Почему? Все просто: автор здесь подчеркивает две последние буквы аббревиатуры RMA как Manufacturing Approval, то есть одобрение производства, и именно производственные процессы, предприятие и экосистема гарантии качества должны быть предварительно одобрены. Поэтому покупатель данных такого одобрения, даже по лицензии, обязан получить собственное одобрение изготовителя частей. Правда, это должно быть проще при приобретении лицензированной конструкции, которая раньше уже имела собственное одобрение. Дополнительные примечания относительно такого одобрения кратко изложены ниже:

Ключевые примечания относительно одобрения изготовителя частей (Parts Manufacturer Approval, PMA)

- Применяется для изготовления частей, влияющих на безопасность и устанавливаемых на воздушное судно.
- Относится к конкретному устройству, а также к конкретным процессам и процедурам для него.
- Держатели одобрения изготовителя частей могут изготавливать части, даже если они не были исходным изготовителем.
- Можно производить модификационные части по дополнительному сертификату типа (Supplemental Type Certificate, STC); одобрение изготовителя частей обычно следует за таким сертификатом.
- Приказы Федерального управления гражданской авиации США (FAA) 8110.42 и 8120.22 описывают процедуры оценки, одобрения и управления производственной деятельностью изготовителя.

Таким образом, одобрение изготовителя частей образует важную часть пирамиды летной годности. С 2011 года больше внимания уделяется системе качества изготовителя; сертифицирующие органы, такие как FAA, признают это и регулярно проводят аудит соответствия в рамках постоянного процесса управления сертификацией.

Заключение

Пирамида летной годности дает простой способ понять на первый взгляд разрозненные аспекты типовой сертификации авионики. Подробности разбросаны по тысячам страниц применимых правил, стандартов, руководящих материалов, норм и приказов. Если следовать описанному здесь пирамидальному подходу слой за слоем, сертификация оказывается не сложнее подъема по высокой изогнутой лестнице: нужны только время, силы, смелость и настойчивость. Наслаждайтесь подъемом; вид сверху действительно выдающийся.

ARP4761A и безопасность в авиации

Автор: Вэнс Хилдерман

При участии: г-н Джон Линч

Рецензент:

Nazan Gozay Gurbuz, эксперт по безопасности и член комитета ARP4754A/4761A

4761, 4754... магические числа или реальные ответы?

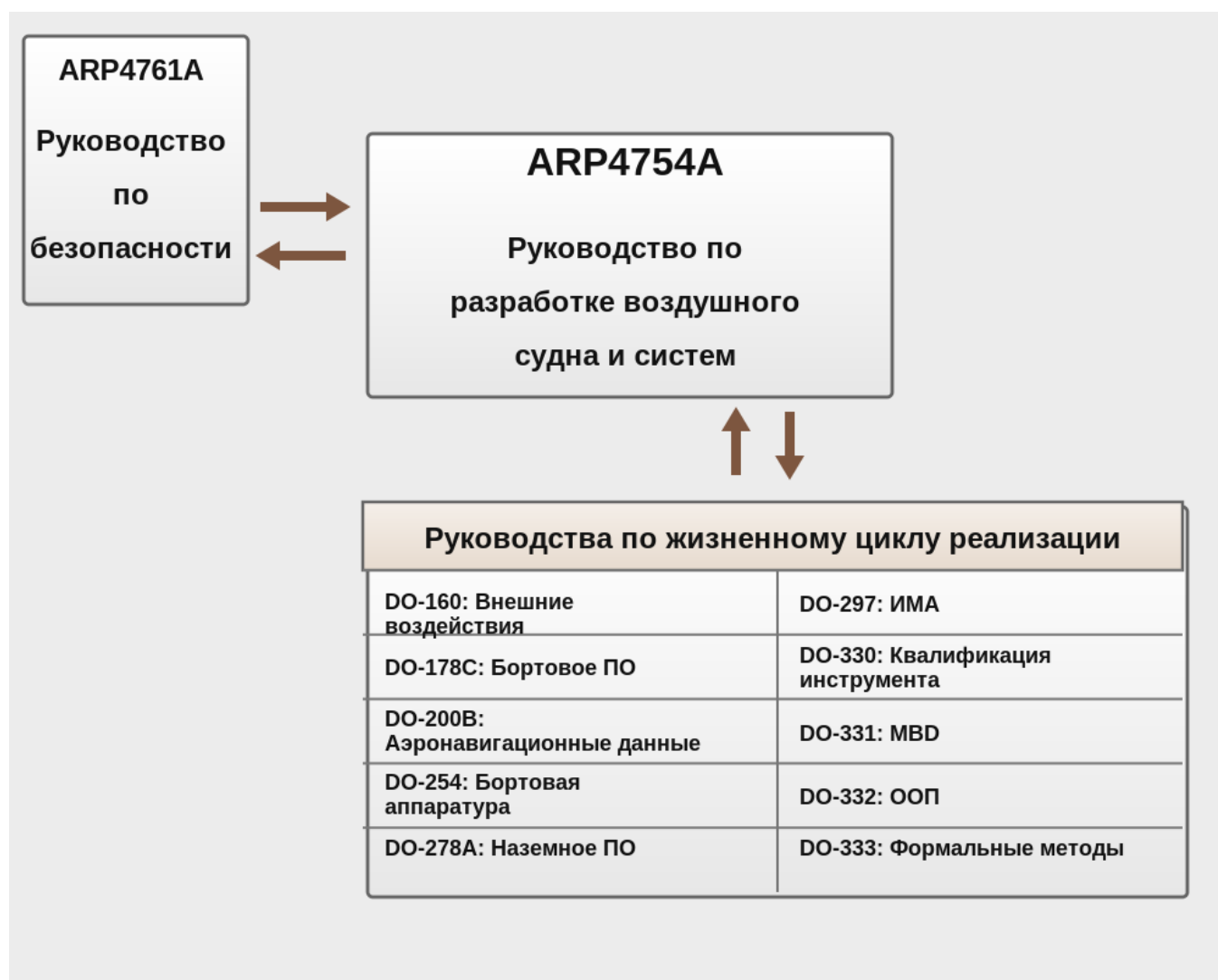
Безопасность полетов не держится на магических числах. Она начинается с поиска реальных ответов. И она не возникает случайно: настоящая, математически обоснованная безопасность должна предотвращать авиационные происшествия. Числа 4754 и 4761 не магические, но напрямую связаны с безопасностью полетов. Подход к безопасности и сами эти документы развивались; новые ответы для авиационной разработки и обеспечения безопасности содержатся в 4754A и 4761A, то есть в документах SAE: ARP4754A по разработке воздушных судов и систем и ARP4761A по оценкам безопасности.

Старые времена.

В технике “старыми временами” можно считать любое время до прошлого года. В очень старые времена – высадки на Луну, космические шаттлы, новые четырехдвигательные коммерческие самолеты – безопасность обеспечивали инженерной смекалкой и доводкой: сильные инженеры делали все возможное, чтобы предотвратить происшествия, а затем конструкцию дорабатывали там, где этих усилий оказывалось недостаточно. Вычислительных ресурсов было меньше, графики были свободнее, и приемлемый уровень безопасности в целом достигался: у космических шаттлов была вероятность не взорваться 98,5 %, а коммерческие самолеты терпели катастрофы со смертельным исходом “всего лишь” несколько раз в год. Затем очень старые времена сменились просто старыми, и прежние методы стали работать хуже. Одни говорили, что инженеры уже не так умны; другие – что эти инженеры пытаются сделать компьютеры слишком умными. Но все соглашались: безопасность нужно формализовать сильнее, а SAE уже десятилетиями выпускала аэрокосмические рекомендуемые практики (Aerospace Recommended Practice, ARP). SAE ARP4754, *Руководство по разработке гражданских воздушных судов и систем*, был опубликован в ноябре 1996 года. Его тесно связанный “собрат” ARP4761, *Руководство и методы проведения процесса оценки безопасности гражданских бортовых систем и оборудования*, вышел через месяц, в декабре 1996 года.

ARP4761A – это больше, чем руководство по безопасности воздушных судов. Его официальное название после выпуска в 2018 году – “Руководство и методы проведения процесса оценки безопасности гражданских бортовых систем и оборудования”. По сути, ARP4761 почти учебное пособие по общей безопасности и применению различных методов анализа для оценки текущих работ по разработке с точки зрения безопасности воздушного судна.

ARP4761A тесно связан с ARP4754A и задает основу для самого фундаментального аспекта авиационных норм – безопасности. Если посмотреть на экосистему авиационной разработки и сертификации, место ARP4761A в верхнем левом углу ясно показывает его важность:



Процесс оценки безопасности – жизненно важный элемент безопасности полетов. Для воздушных судов и авионики его основу задает ARP4761 и его новая версия ARP4761A.

Практически каждый аспект авиации проходит оценку безопасности: нужно понять потенциальные риски, количественно их оценить, а затем предотвратить, обнаружить или снизить. Опытные специалисты авиационной отрасли не преувеличивают, когда говорят, что процесс оценки безопасности, возможно, самый важный элемент разработки авионики. Для авионики его роль состоит в обеспечении безопасности воздушного судна,

экипажа и находящихся на борту лиц. По существу, безопасность воздушного судна повышается за счет тщательного анализа, оптимизации архитектуры, определения уровня критичности, выбора компонентов, доработки архитектуры, мониторинга и технического обслуживания. Только полноценный процесс оценки безопасности позволяет получить архитектуру с дополнительными требованиями, связанными с безопасностью. Название ARP4761 хорошо объясняет его значение в этом фундаментальном процессе:

“Руководство и методы проведения процесса оценки безопасности гражданских бортовых систем и оборудования”

В идеальном мире оценки безопасности авионики выполнялись бы по универсальному набору единых шагов анализа. В том же идеальном мире люди никогда бы не болели, не набирали вес и не старели. К сожалению, идеальный мир гарантировать нельзя. Отсюда базовая предпосылка оценки безопасности: анализировать потенциальные проблемы нужно для реальных, а не идеальных условий эксплуатации. Поэтому оценки безопасности жизненно важны. В авиации идеального мира нет, и разработчики критичных систем авионики тратят огромное количество времени на бесчисленные сценарии “что если” и наихудшие случаи, связанные с потенциальными и уже проявившимися неисправностями.

Если бы все разработчики авионики были безупречными гениями и могли в уме сформулировать и решить все возможные сценарии безопасности, им не понадобился бы письменный формальный процесс оценки безопасности. Они ведь уже все сделали бы идеально. Но кто бы это рассмотрел? Как обеспечить согласованность между оценками? Что делать с будущими изменениями систем? Можно ли называть процессом то, что нигде не определено? И кто на практике является непогрешимым гением? Ответы очевидны, а значит, очевидна и необходимость формального процесса оценки безопасности. ARP4761 задает основу такого процесса, но при объеме более 300 страниц в этом руководстве здесь возможен только краткий обзор. Чтобы понимать и применять ARP4761, специалисту нужно видеть общий контекст безопасности и связь с другими документами.

Безопасность полетов закреплена в большой экосистеме информационных материалов и нормативных требований. Каждый, кто даже косвенно участвует в обеспечении безопасности полетов, действует по правилам, которые как минимум требуют определить процедуры и следовать им. В большинстве случаев дополнительные нормативные документы задают правила или рекомендации о том, как выполнять задачи и какой минимальный порог нужно достигнуть для сертификации. К областям авиации, регулируемым требованиями, связанными с безопасностью, относятся:

- пилоты и экипажи воздушных судов;
- наземные операторы;
- персонал технического обслуживания;
- управление воздушным движением, связь, навигация и т. д.;
- разработка систем, аппаратуры и ПО, включая бортовые и наземные средства;

- инструменты, включая механические, аппаратные и программные;
- практически все аспекты производства воздушных судов;
- летные испытания;
- поставщики, производство и обновления.

Для авионики набор нормативных документов более сфокусирован. Процесс оценки безопасности авионики в основном опирается на следующие отраслевые и государственные документы:

- **SAE ARP4761/A:** *Руководство и методы проведения процесса оценки безопасности гражданских бортовых систем и оборудования*
- **SAE ARP4754A:** *Руководство по разработке гражданских воздушных судов и систем*
- **Консультативный циркуляр Федерального управления гражданской авиации США (FAA) AC23.1309-1E:** *Оборудование, системы и установки в самолетах части 23 (14 CFR Part 23)*
- **Консультативный циркуляр FAA AC25.1309-1A:** *Проектирование и анализ систем*
- **Консультативный циркуляр FAA AC27-1B:** *Сертификация винтокрылых аппаратов нормальной категории*
- **Консультативный циркуляр FAA AC29-2C:** *Сертификация винтокрылых аппаратов транспортной категории*
- **Радиотехническая комиссия по авиации (Radio Technical Commission for Aeronautics, RTCA) DO-160:** *Условия окружающей среды и процедуры испытаний бортового оборудования*
- **RTCA DO-178C:** *Вопросы ПО при сертификации бортовых систем и оборудования*
- **RTCA DO-254:** *Руководство по гарантии разработки бортовой электронной аппаратуры*

Если бы эта глава была посвящена безопасности полетов в целом или даже безопасности воздушного судна в целом, охват был бы намного шире. Но здесь цель уже: отдельные системы авионики. В процессе оценки безопасности участвуют опытные инженеры, знакомые с указанными выше документами и нормативными требованиями; они выполняют серию анализов, относящихся к системе и воздушному судну. По мере выполнения оценки безопасности эти анализы обычно сопровождаются следующей документацией:

- *Оценка функциональных опасностей воздушного судна (Aircraft Functional Hazard Assessment, AFHA);*
- *Анализ дерева отказов воздушного судна (Aircraft Fault Tree Analysis, FTA);*
- *Оценка функциональных опасностей системы (System Functional Hazard Assessment, SFHA);*
- *Анализ дерева отказов системы (System Fault Tree Analysis, FTA);*
- *Анализ видов и последствий отказов системы (System Failure Modes and Effects Analysis, FMEA), анализ дерева отказов изделия (Item Fault Tree Analysis, FTA);*

- *Анализ видов и последствий отказов изделия (Item Failure Modes and Effects Analysis, FMEA).*

Ключевое понятие здесь – “трассируемость”. Связи между эксплуатацией воздушного судна, техническим обслуживанием, системами и другими областями нужно формализовать и связать между собой; это и называется трассируемостью. Раньше ее можно было вести вручную или с помощью крупных корпоративных инструментов с большими базами данных. Но ручные подходы и такие тяжелые инструменты плохо показывали сами связи, оценки безопасности, развитие документации, свидетельства рассмотрения и производные требования. Под производными здесь понимаются требования, которые выводятся из оценки безопасности, добавляются для уточнения проектного или реализационного решения либо закрывают пробел.

Безопасность полетов включает четыре ключевых вида деятельности, показанные на рисунке ниже:

Четыре ключевых вида деятельности по безопасности

1. Оценка функциональных опасностей (Functional Hazard Assessment, FHA): выявить потенциальные отказы и их последствия, затем классифицировать тяжесть каждого отказа.
2. Предварительная оценка безопасности воздушного судна (Preliminary Aircraft Safety Assessment, PASA) или предварительная оценка безопасности системы (Preliminary System Safety Assessment, PSSA): проанализировать предложенную архитектуру, определить, как могут возникать отказы, выявленные при оценке функциональных опасностей, и сформировать требования безопасности.
3. Оценка безопасности воздушного судна или системы: оценить системы воздушного судна и проверить выполнение требований безопасности.
4. Анализ общих причин (Common Cause Analysis, CCA): проверить, достаточно ли независимы функции и системы для достижения заданного уровня безопасности.

У области безопасности есть собственный язык и словарь. Опытные инженеры по безопасности хорошо им владеют, но, вероятно, вы не читали бы эти строки, если бы уже были опытным инженером по безопасности авионики. Для начала подумайте над следующими словами и проверьте себя: сформулируйте вслух определение каждого термина в 1-2 предложениях. Ответы приведены далее, но сначала стоит потратить несколько минут на самостоятельное размышление. Водить автомобиль тоже не учатся одним чтением руководства пользователя; настоящее обучение начинается на практике.

- *Неблагоприятное воздействие (Adverse Effect)? Оценка (Assessment)?*
- *Средняя вероятность на летный час (Average probability per flight hour)?*
- *Сложная система (Complex System)?*

- *Уровень гарантии разработки (Development Assurance Level, DAL)?*
- *Крайне маловероятное отказное состояние (Extremely remote failure condition)?*
- *Практически невероятное отказное состояние (Extremely improbable failure condition)?*
- *Отказное состояние (Failure condition)?*
- *Незначительное (Minor)?*
- *Значительное (Major)?*
- *Опасное (Hazardous)?*
- *Катастрофическое (Catastrophic)?*
- *Оценка функциональных опасностей (Functional Hazard Assessment)?*
- *Опасность (Hazard)?*
- *Основная функция (Primary Function)?*
- *Основная система (Primary System)? Надежность (Reliability)?*
- *Резервная система (Secondary System)? Простая система (Simple System)? Система (System)?*

Итак, вы подумали над этими словами. Что они означают применительно к безопасности авионики? Во-первых, нет обязательного определения, которому следовали бы все. Если посмотреть десяток самых распространенных источников авиационной терминологии, обнаружится множество небольших различий в определениях этих терминов, причем некоторые различия окажутся совсем не мелкими. Разные органы и авторы стараются поддерживать согласованность; в авиации это называют “гармонизацией”, например когда Федеральное управление гражданской авиации США и Агентство Европейского союза по авиационной безопасности (European Union Aviation Safety Agency, EASA) координируют работу для устранения различий. Но реальный мир постоянно развивается, и его словарь непрерывно расширяется. По совпадающему по названию, но не связанному с этим принципу Гейзенберга не стоит ожидать идеального консенсуса по всем определениям. Поэтому прежде чем использовать приведенные далее определения, изучите обязательные нормы, применимые к вашей деятельности по авионике, и согласуйте их с органами одобрения, указав источник и само определение в документации по авионике. После этих настороженно-мудрых слов (wary wise words, WWW) ниже приведены определения, которые автор считает наиболее общепринятыми:

Оценка (Assessment): оценка, основанная на инженерном суждении.

Средняя вероятность на летный час (Average probability per flight hour): представление числа случаев, когда прогнозируется возникновение рассматриваемого отказного состояния в течение всего эксплуатационного срока всех самолетов данного типа, деленное на ожидаемое суммарное количество летных часов всех самолетов этого типа.

Сложная система (Complex System): система, работа которой, виды отказов или последствия отказов трудно понять без помощи аналитических методов или структурированных методов оценки.

Уровень гарантии разработки (Development Assurance Level, DAL): все запланированные и систематические действия, используемые для подтверждения с достаточной уверенностью, что ошибки в требованиях, проекте и реализации выявлены и исправлены, а системы и изделия (аппаратура и ПО) удовлетворяют применимому сертификационному базису. Пять уровней гарантии разработки обозначаются буквами А, В, С, D и Е, где А – наивысший, а Е – самый низкий.

Крайне маловероятное отказное состояние (Extremely remote failure condition): такие отказные состояния, возникновение которых не ожидается для каждого отдельного самолета в течение всего срока его службы, но которые могут случиться несколько раз, если рассматривать суммарный эксплуатационный срок всех самолетов данного типа.

Практически невероятное отказное состояние (Extremely improbable failure condition): для самолетов пригородной категории – такие отказные состояния, вероятность которых столь мала, что их возникновения не ожидается в течение всего эксплуатационного срока всех самолетов одного типа. Для других классов самолетов вероятность возникновения может быть выше.

Отказное состояние (Failure condition): состояние, оказывающее влияние либо на самолет, либо на находящиеся на борту лиц, либо на тех и других, непосредственно или косвенно, которое вызвано одним или несколькими отказами или ошибками либо которому они способствуют, с учетом фазы полета и соответствующих неблагоприятных эксплуатационных или внешних условий либо внешних событий. Отказные состояния могут классифицироваться по тяжести следующим образом:

Без влияния на безопасность (No Safety Effect): отказные состояния, которые не оказывают влияния на безопасность (то есть такие отказные состояния, которые не влияют на эксплуатационные возможности самолета и не увеличивают нагрузку на экипаж).

Незначительное (Minor): отказные состояния, которые не приводили бы к существенному снижению безопасности самолета и требовали бы от экипажа действий в пределах его возможностей. Они могут включать небольшое снижение запасов безопасности или функциональных возможностей, небольшое увеличение нагрузки на экипаж (например, обычные изменения плана полета) либо некоторый физический дискомфорт для пассажиров или кабинного экипажа.

Значительное (Major): отказные состояния, которые снижали бы возможности самолета или способность экипажа справляться с неблагоприятными условиями эксплуатации настолько, что возникало бы существенное снижение запасов безопасности или функциональных возможностей. Кроме того, такое отказное состояние сопровождается значительным увеличением нагрузки на экипаж, условиями,

снижающими эффективность экипажа, дискомфортом для летного экипажа либо физическими страданиями пассажиров или кабинного экипажа, возможно включая травмы.

Опасное (Hazardous): отказные состояния, которые снижали бы возможности самолета или способность экипажа справляться с неблагоприятными условиями эксплуатации настолько, что возникало бы любое из следующего:

- a. значительное снижение запасов безопасности или функциональных возможностей;
- b. физические страдания или более высокая нагрузка, из-за которых на летный экипаж нельзя было бы полагаться в точном и полном выполнении своих задач; или
- c. серьезная травма или смертельное поражение лица на борту, не являющегося членом летного экипажа.

Катастрофическое (Catastrophic): отказные состояния, которые, как ожидается, приводят к множественным смертельным исходам среди находящихся на борту лиц либо к потере работоспособности или смертельной травме члена летного экипажа, как правило с потерей воздушного судна.

Оценка функциональных опасностей (Functional Hazard Assessment): систематическое, всестороннее рассмотрение функций для выявления и классификации отказных состояний этих функций по степени их тяжести.

Опасность (Hazard): состояние, возникающее вследствие отказов, внешних событий, ошибок или их сочетаний, при котором затрагивается безопасность.

Надежность (Reliability): вероятность того, что система или изделие будет выполнять требуемую функцию в заданных условиях, без отказа, в течение заданного периода времени.

Простая система (Simple System): система, для которой можно полностью обеспечить гарантии (валидацию и верификацию) сочетанием испытаний и анализа по отношению к ее требованиям и выявленным отказным состояниям.

Система (System): совокупность взаимосвязанных изделий, организованных для выполнения определенной функции или функций.

Распространенное заблуждение состоит в том, что цель оценки безопасности – полностью устранить саму возможность опасностей. Цель благородная, но для сложных асинхронных систем авионики невозможная. Вместо этого потенциальные опасности нужно выявлять и количественно оценивать согласно следующему принципу:

- *серьезная опасность может быть допустимой, если вероятность ее возникновения приемлемо мала;*
- *вероятная опасность может быть допустимой, если воздействие опасности на воздушное судно, летный экипаж и находящихся на борту лиц является приемлемым;*

- вероятность возникновения опасности должна быть обратно пропорциональна степени ее тяжести.

Связь между тяжестью отказных состояний и требованиями по безопасности обобщена в следующей таблице.

**Тяжесть отказных состояний и требования по безопасности
(перепечатано из ARP 4761)**

	НЕЗНАЧИТЕЛЬНОЕ	ЗНАЧИТЕЛЬНОЕ	ТЯЖЕЛОЕ ЗНАЧИТЕЛЬНОЕ / ОПАСНОЕ	КАТАСТРОФИЧЕСКОЕ
Воздействие на воздушное судно	Небольшое снижение функциональных возможностей или запасов безопасности	Существенное снижение функциональных возможностей или запасов безопасности	Большое снижение функциональных возможностей или запасов безопасности	Делает невозможными продолжение безопасного полета и посадку
Воздействие на летный экипаж	Небольшое увеличение нагрузки или использование аварийных процедур	Физический дискомфорт либо значительное увеличение нагрузки	Физические страдания или чрезмерная нагрузка, ухудшающая способность выполнять задачи	Смертельные исходы или потеря работоспособности
Воздействие на находящихся на борту лиц	Физический дискомфорт	Физические страдания, возможно включая травмы	Серьезная или смертельная травма небольшого числа лиц на борту	Множественные смертельные исходы
Требование по вероятности	Вероятно < 1.0E-3 на летный час	Маловероятно < 1.0E-5 на летный час	Крайне маловероятно < 1.0E-7 на летный час	Практически невероятно < 1.0E-9 на летный час
Уровень гарантии разработки	D	C	B	A

Риски, связанные с отказами, снижают за счет качественных и/или количественных требований по безопасности, как показано в таблице выше. Ошибки же снижают за счет реализации процесса гарантии разработки, от DAL A до DAL E. Целостность относительно систематических отказов достигается именно средствами процесса гарантии разработки.

Типичные шаги процесса безопасности для соответствия ARP4761A и ARP4754A в этих областях таковы:

- идентификация отказных состояний на уровне воздушного судна и на уровне систем;

- идентификация и/или выбор мер снижения риска на уровне воздушного судна или системы либо эксплуатационных ограничений;
- разработка требований по безопасности для фиксации мер снижения риска отказных состояний;
- распределение требований по безопасности, связанных с этими мерами, по воздушному судну и/или системам.

Каждый из этих шагов подробнее описан ниже. Отдельно снова рекомендуем Honda вести внутреннюю историю развития документации и записи рассмотрений по всем этим материалам, чтобы позднее, при аудите FAA, можно было показать фактическое соблюдение требований на протяжении всей разработки в соответствии с основными руководствами ARP4754A/ARP4761.

Идентификация отказных состояний на уровне воздушного судна и на уровне систем

Следует разработать оценку функциональных опасностей (Functional Hazard Assessment, FHA) на уровне воздушного судна и оценки функциональных опасностей на уровне систем. Как минимум, они должны выявлять отказные состояния потери функции и отказные состояния неправильного функционирования. Отказные состояния потери функции иногда называют отказными состояниями доступности. Отказные состояния неправильного функционирования иногда называют отказными состояниями целостности. Полезные пояснения по доступности и целостности приведены в ARP4754A – сопутствующем документе к ARP4761. ARP4754A рассматривается в следующей главе, но кратко эти определения приведены ниже:

ARP4754A определяет доступность следующим образом:

- качественный или количественный атрибут, показывающий, что система или изделие находится в работоспособном состоянии в заданный момент времени. Иногда доступность выражают через вероятность того, что система или изделие не выдает свои выходные данные, то есть через недоступность. [Примечание AFuzion: недоступность эквивалентна потере функции.]

ARP4754A определяет целостность следующим образом:

- качественный или количественный атрибут системы или изделия, показывающий, что на его правильную работу можно положиться. Иногда целостность выражают через вероятность невыполнения критериев правильной работы. [Примечание AFuzion: невыполнение критериев правильной работы эквивалентно неправильному функционированию.]

Некоторые типы систем используют и другие виды классификации отказных состояний, помимо доступности и целостности. Например, оценки функциональных опасностей, связанные с функцией требуемых навигационных характеристик (Required Navigation Performance, RNP), обычно выявляют отказные состояния, связанные с непрерывностью. Непрерывность определяется так:

- способность всей системы выполнять свою функцию без перерыва в ходе предполагаемой операции. Опасность при ухудшении непрерывности состоит в том, что работа системы прервется и она не предоставит информацию наведения для предполагаемой фазы полета.

ФНА на уровне воздушного судна должна выявлять все предвидимые отказные состояния потери функции (доступности) и неправильного функционирования (целостности). ФНА на уровне системы должна выявлять все предвидимые отказные состояния доступности и целостности для конкретной оцениваемой системы. Если ФНА системы выявляет отказное состояние доступности или целостности, которого еще нет в ФНА воздушного судна, такое отказное состояние нужно поднять на уровень ФНА воздушного судна, чтобы обеспечить ее полноту. ФНА системы связана с ФНА воздушного судна через распределение функций воздушного судна на функции системы. Отказные состояния из ФНА воздушного судна находятся в отношении “родитель-потомок” к отказным состояниям из ФНА системы. Так обеспечивается нисходящая трассируемость.

Идентификация и/или выбор мер снижения риска на уровне воздушного судна или системы

Часто в конструкцию воздушного судна и/или системы необходимо включать меры снижения риска, чтобы уменьшить вероятность возникновения определенных отказных состояний до приемлемого уровня. Ниже приведен неполный перечень таких мер:

- строгость процесса разработки в соответствии с назначенным уровнем гарантии разработки функции (Function Development Assurance Level, FDAL) или уровнем гарантии разработки элемента (Item Development Assurance Level, IDAL);
- сигнализация летному экипажу о потере функции или неправильном функционировании;
- обнаружение, изоляция и исключение неисправности;
- реакция на неисправность;
- резервная функциональность;
- резервирование;
- использование высоконадежных (то есть с низкой интенсивностью отказов) компонентов;
- независимость;
- обособление и защита;
- неоднородность и разнообразие компонентов;

- физическое обособление компонентов;
- электрическая изоляция между компонентами;
- средства мониторинга (например, диапазона, скорости изменения, устойчивости);
- ограничители;
- компараторы;
- предохранительные блокировки;
- обработчики исключений;
- циклические избыточные коды (Cyclic Redundancy Code, CRC) и контрольные суммы;
- обнаружение и исправление ошибок;
- сторожевые таймеры;
- встроенный контроль (Built-In Test, BIT): при включении, непрерывный, иницилируемый.

Оценки функциональных опасностей и предварительные оценки безопасности обычно выявляют меры снижения риска там, где они необходимы.

Разработка требований по безопасности для фиксации мер снижения риска отказных состояний

Предварительные оценки безопасности воздушного судна (PASA) и предварительные оценки безопасности системы (PSSA) обычно используются для фиксации требований по безопасности. При разработке PASA и PSSA эти требования определяются так, чтобы обеспечить реализацию мер снижения риска отказных состояний. Все требования по безопасности, включая требования, относящиеся к таким мерам, включаются в общий набор требований.

Распределение требований по безопасности, связанных с мерами снижения риска, по воздушному судну и/или системам

Следующий шаг процесса разработки воздушного судна и систем – распределить требования, включая требования по безопасности, относящиеся к мерам снижения риска, на воздушное судно, системы или изделия. Требования также можно распределять на другие элементы, например:

- производственные требования;
- требования к техническому обслуживанию;
- требования к калибровке;
- эксплуатационные ограничения или ограничительные условия;
- подготовка летного экипажа.

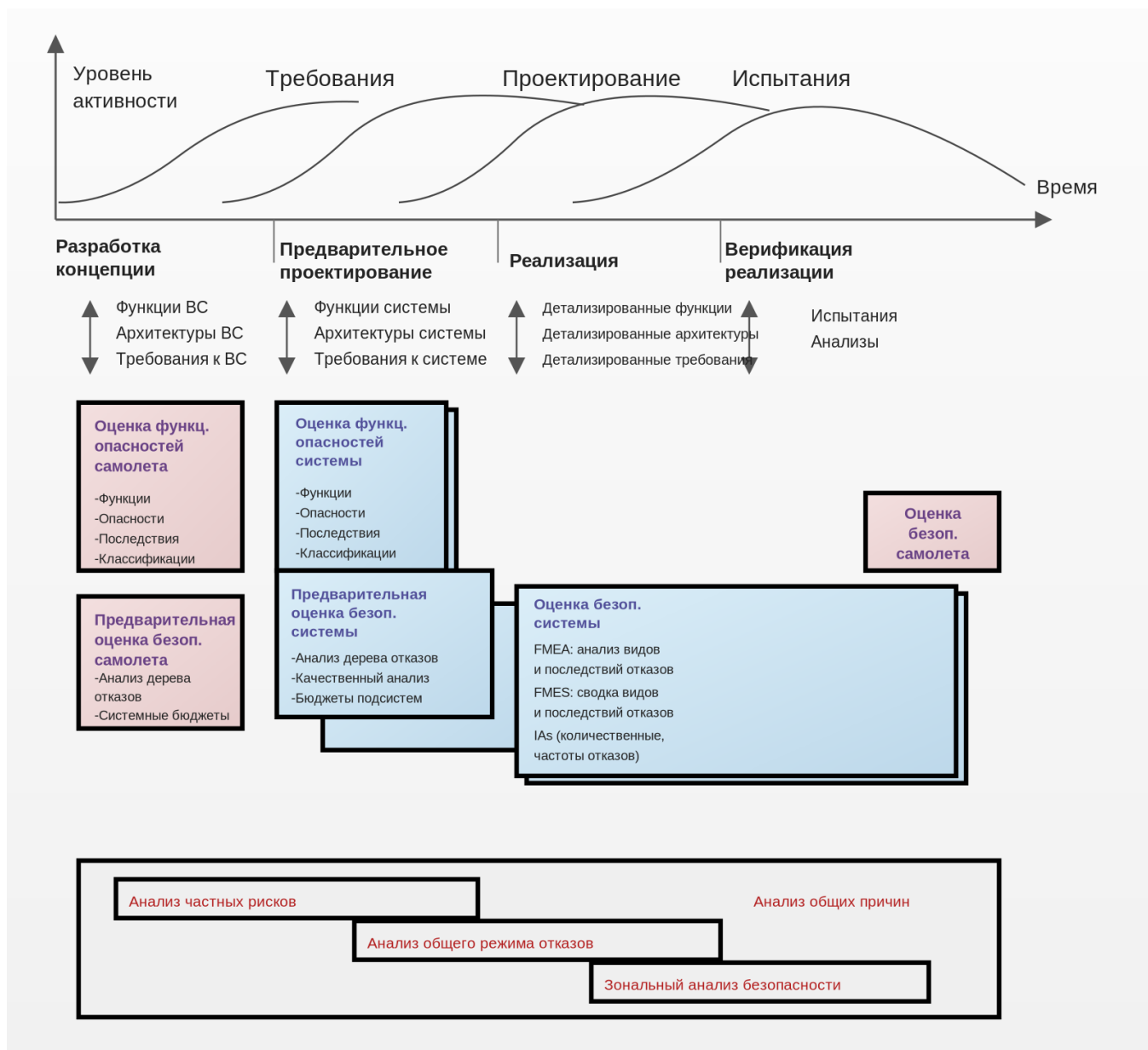
Требование следует распределять на элемент только тогда, когда этот элемент действительно способен его реализовать. Если требование назначено элементу, который не может его выполнить, нужно зарегистрировать сообщение о проблеме или использовать аналогичный механизм, чтобы решить этот вопрос. Проблему распределения требований можно устранить несколькими способами, например:

- выявить надлежащий элемент распределения (например, перераспределить требование уровня системы на уровень воздушного судна);
- изменить назначенное требование на такое, которое может быть выполнено элементом, которому это требование назначено;
- изменить архитектуру системы таким образом, чтобы назначенное требование могло быть выполнено.

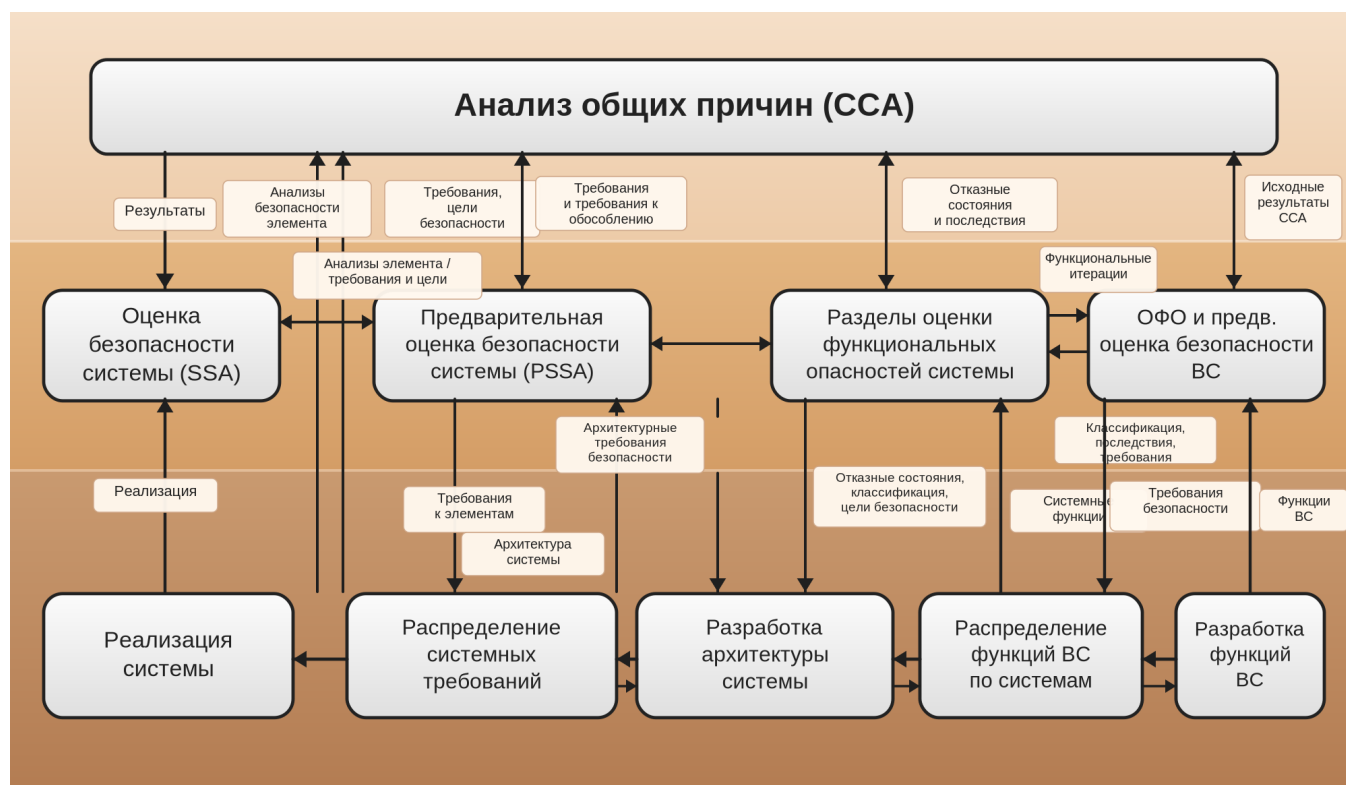
Внутренние отказные состояния (опасности)

Помимо функциональных отказных состояний, воздушное судно или его оборудование могут содержать внутренние, физически присущие им опасности. Внутренние опасности воздушного судна выявляют в ходе зонального анализа безопасности (Zonal Safety Analysis, ZSA), рассматривая возможные взаимные воздействия между системами в общей зоне воздушного судна. Внутренние опасности оборудования выявляют при оценке ведомости материалов сборочной единицы (Bill of Materials, BOM): опасные материалы, тепловой разгон, высокое напряжение, радиацию, химическую токсичность, взрывоопасность и тому подобное.

Следующий рисунок показывает взаимодействие между видами деятельности процесса оценки безопасности и процесса разработки. Процесс оценки безопасности начинается на этапе разработки концепции и заканчивается верификацией того, что проект удовлетворяет требованиям по безопасности. Из-за итеративного характера проектирования в конструкцию вносятся изменения, и измененный проект нужно заново оценивать с точки зрения безопасности. “Итеративный подход” из ARP4754 был расширен в ARP4754A совместно с ARP4761, чтобы разработчики применяли формальный итерационный процесс в области безопасности и непрерывно уточняли оценки безопасности на протяжении всего проекта.



Последовательность перечисленных выше видов деятельности по безопасности показана на следующем рисунке. Обратите внимание: рисунок начинается в правом нижнем углу и движется вверх и влево:



Оценка безопасности должна отвечать на следующие вопросы сначала для воздушного судна, а затем для каждой системы:

Ключевые вопросы оценки безопасности для воздушного судна и систем

1. Что выполняет рассматриваемая функция или система?
2. Что может пойти не так?
3. Что произойдет при отказе?
4. Что может вызвать отказ?
5. Каков риск?
6. Можно ли принять этот риск?

Важно помнить: процесс оценки безопасности начинается с общей оценки функциональных опасностей, затем следует предварительная оценка; обе выполняются сверху вниз. Далее, во время реализации и после нее, выполняется окончательная оценка безопасности системы, включая анализ видов и последствий отказов, который проводится снизу вверх. Этот процесс показан ниже:

Последовательность оценок безопасности сверху вниз и снизу вверх

Три ключевые оценки безопасности выполняются сначала сверху вниз, затем снизу вверх; по мере уточнения проекта цикл повторяется.

1. Оценка функциональных опасностей (Functional Hazard Assessment, FHA).
2. Предварительная оценка безопасности системы (Preliminary System Safety Assessment, PSSA).
3. Оценка безопасности системы (System Safety Assessment, SSA).

Оценка функциональных опасностей (FHA)

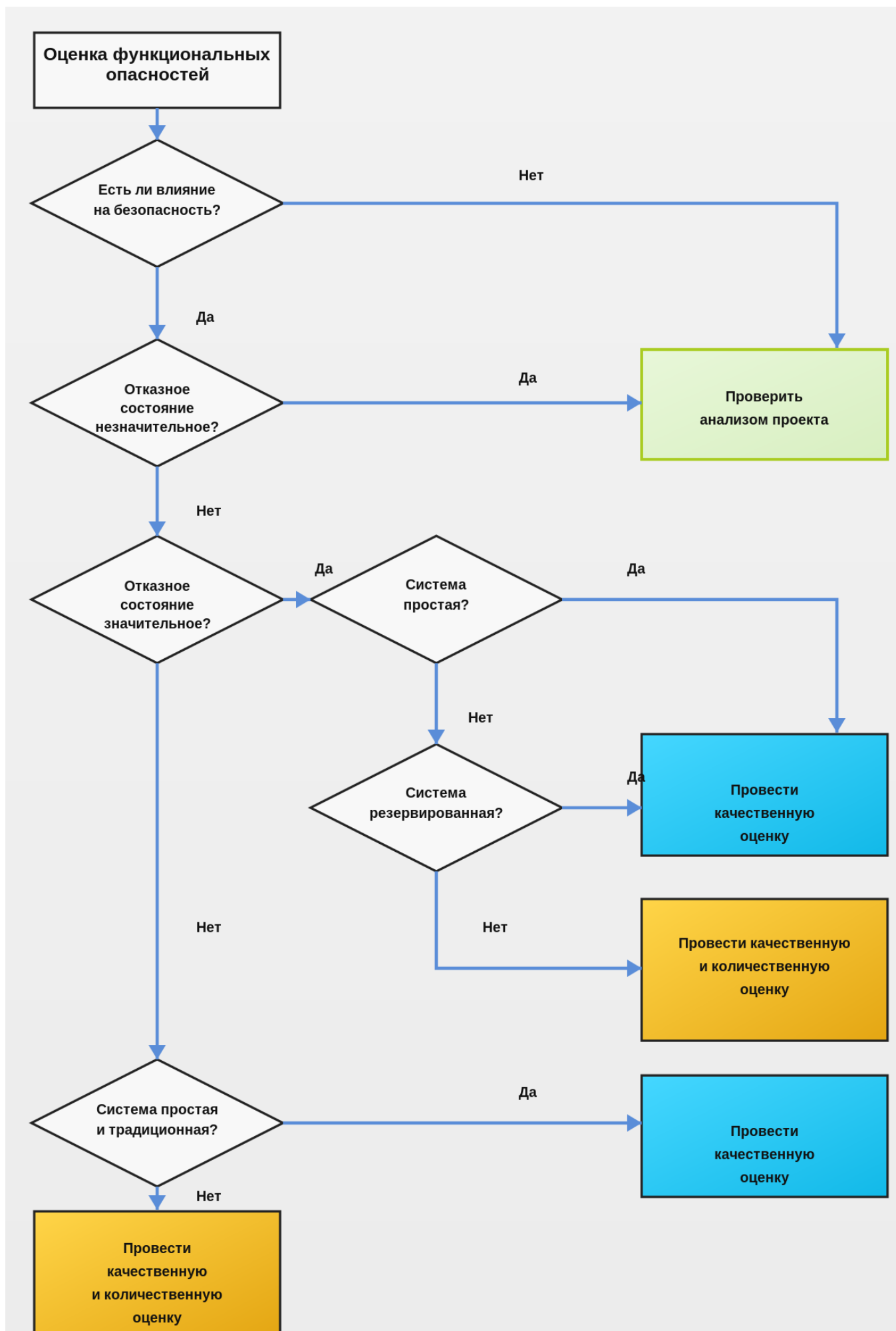
Оценки функциональных опасностей (FHA) выполняются как на уровне воздушного судна, так и на уровне систем. Их цель – выявить отказные состояния функций воздушного судна и систем (потерю функции, неправильное функционирование и т. д.) и классифицировать их по тяжести (катастрофические, опасные, значительные и т. д.), чтобы можно было предложить и реализовать конструкции воздушного судна и систем, снижающие вероятность этих отказных состояний до приемлемо низкого уровня. При сертификации авионики все стороны признают важность FHA. Заявитель отвечает за выявление каждого отказного состояния и выбор методов оценки безопасности. Затем ему следует заранее согласовать с компетентным сертифицирующим органом идентификацию отказных состояний, их классификацию и выбор приемлемого метода определения соответствия.

Назначение оценки функциональных опасностей кратко показано на рисунке ниже:

Назначение оценки функциональных опасностей (Functional Hazard Assessment, FHA)

1. Определить все функции воздушного судна или системы.
2. Выявить и описать все отказные состояния.
3. Определить последствия каждого отказного состояния.
4. Классифицировать каждое отказное состояние.
5. Определить требования, реализующие меры снижения риска для каждого отказного состояния.
6. Определить подтверждающие материалы, обосновывающие выбранные меры снижения риска.
7. Определить методы верификации для каждого требования.
8. Определить уровень гарантии разработки функции (Function Development Assurance Level, FDAL).

На следующем рисунке приведен обзор информационных потоков процесса оценки безопасности.



Порядок выполнения оценки функциональных опасностей (ФНА)

1. Прежде чем заявитель перейдет к детальной оценке безопасности, следует подготовить ФНА функций самолета и систем, чтобы определить необходимость и объем последующего анализа. Такая оценка может опираться на опыт эксплуатации, инженерное и эксплуатационное суждение либо на сочетание опыта эксплуатации с нисходящим дедуктивным качественным рассмотрением каждой функции. ФНА представляет собой систематическое, всестороннее рассмотрение функций самолета и систем для выявления потенциальных отказных состояний без влияния на безопасность, а также незначительных, значительных, опасных и катастрофических отказных состояний. Они могут возникать не только из-за неправильного функционирования или отказа выполнять функцию, но и вследствие нормальных реакций на необычные или ненормальные внешние факторы. ФНА рассматривает эксплуатационные уязвимости систем, а не детальный анализ фактической реализации.
2. Каждая функция системы должна рассматриваться с учетом других функций, выполняемых этой системой, поскольку потеря или неправильное функционирование всех функций, выполняемых системой, может привести к более тяжелому отказному состоянию, чем потеря одной функции. Кроме того, каждая функция системы должна рассматриваться с учетом функций, выполняемых другими системами самолета, поскольку потеря или неправильное функционирование различных, но связанных функций, обеспечиваемых отдельными системами, может повлиять на тяжесть отказных состояний, постулируемых для конкретной системы.
3. ФНА – инженерный инструмент, который следует применять на раннем этапе проектирования и обновлять по мере необходимости. Она используется для определения высокоуровневых целей безопасности самолета или системы, которые должны учитываться в предлагаемых архитектурах. Также она помогает определять DAL для систем. Для многих систем заявителю может быть достаточно простого рассмотрения проекта системы, чтобы определить классификацию опасности. Выполнение ФНА требует опытного инженерного суждения и ранней координации между заявителем и сертифицирующим органом.

ФНА – это намного больше, чем длинная цепочка выводов. Она также включает требования уровня воздушного судна и системы, позволяющие достичь целей по вероятности отказов, в том числе:

- *проектные ограничения;*
- *соображения резервирования;*
- *задание и сигнализацию отказных состояний;*
- *предлагаемые действия пилота и экипажа по снижению последствий;*
- *рекомендуемые действия по техническому обслуживанию.*

Выходные данные FHA включают:

- перечень функций;
- рабочий лист FHA (таблицу), показывающий следующее:
- идентификация функции;
- отказные состояния;
- фаза полета;
- воздействие отказного состояния на воздушное судно, летный экипаж и находящихся на борту лиц;
- классификация отказного состояния;
- метод верификации того, что требование по вероятности выполнено;
- ссылка на подтверждающие материалы;
- производные требования для систем нижнего уровня.

Предварительная оценка безопасности воздушного судна (PASA) – новая оценка в ARP4761A

PASA представляет собой систематическое рассмотрение предлагаемой архитектуры воздушного судна, чтобы определить, как отказы могут вызвать отказные состояния, выявленные в FHA воздушного судна. Процесс PASA начинается на этапе начальной разработки архитектуры воздушного судна. Поэтому раннее выявление требований по безопасности уровня воздушного судна, таких как уровень гарантии разработки функции, независимость, вероятностные бюджеты и т. д., помогает снизить риски в процессе разработки системы. PASA особенно важна для оценки отказных состояний уровня воздушного судна, когда в выполнении функции воздушного судна участвуют несколько систем.

Предварительная оценка безопасности системы (PSSA)

PSSA представляет собой набор анализов, обычно выполняемых на этапах требований к системе и требований к изделиям в жизненном цикле воздушного судна. Именно в PSSA оцениваются и определяются предлагаемые архитектуры систем; это позволяет выводить требования по безопасности к системе и изделиям. Входными документами для PSSA являются FHA воздушного судна, предварительный FTA воздушного судна и FHA системы. Документы, формируемые в ходе PSSA, включают обновленные FHA воздушного судна и системы, обновленные FTA воздушного судна и систем, а также предварительные анализы общих причин на уровне системы (Common Cause Analysis, CCA).

PSSA выявляет производные требования по безопасности системы, такие как резервирование, обособление, мониторинг, неоднородность и т. д. PSSA также выявляет необходимые уровни гарантии разработки для функций системы и изделий.

PSSA – непрерывный и итеративный процесс, выполняемый на нескольких этапах разработки системы. Цель PSSA – определить, как отказы воздушного судна и системы могут привести к опасностям, выявленным в FHA, и как можно выполнить требования FHA.

Оценка безопасности системы (SSA)

SSA верифицирует, что реализованные конструкции воздушного судна и систем удовлетворяют требованиям FHA воздушного судна и системы, а также требованиям PSSA. Документация, формируемая в ходе SSA, включает:

- *перечень отказных состояний из FHA системы и требований по безопасности, распределенных в SSA вместе с FTA;*
- *результаты качественных оценок каждого отказного состояния или требования по безопасности;*
- *любые материалы, использованные для валидации классификаций отказных состояний;*
- *любые материалы, показывающие, что предположения, использованные при оценке, являются действительными, например процедуры руководства по летной эксплуатации, предельно допустимые значения и т. д.;*
- *документацию, показывающую требования к установке изделий;*
- *при необходимости – пересмотренные руководства по техническому обслуживанию, описывающие новые задачи обслуживания, направленные на сокращение времени пребывания компонентов в опасном состоянии;*
- *при необходимости – пересмотренные руководства по эксплуатации для летного экипажа, описывающие процедуры, которым следует следовать при возникновении определенных отказных состояний;*
- *информацию, показывающую, что система и изделия были разработаны в соответствии с назначенными уровнями гарантии разработки.*

Методы, применяемые для оценки безопасности, показаны на следующем рисунке:

Методы, применяемые для оценки безопасности

- Анализ видов и последствий отказов (Failure Mode and Effects Analysis, FMEA).
- Анализ общего режима отказов.
- Анализ специфического риска (Particular Risk Analysis, PRA).
- Зональный анализ безопасности (Zonal Safety Analysis, ZSA).

Наиболее распространенные методы, применяемые для повышения безопасности воздушного судна и системы за счет улучшений конструкции, непосредственно поддерживающих повышение безопасности и надежности, показаны на следующем рисунке:

Методы повышения безопасности и надежности конструкции

- Резервирование.
- Логическое и физическое обособление, изоляция.
- Разнородность.
- Мониторинг.
- Механизмы защиты и реконфигурации.

Оценка безопасности воздушного судна (ASA) – новая оценка в ARP4761A

Оценка безопасности воздушного судна представляет собой систематическую, всестороннюю оценку реализованного воздушного судна для верификации того, что реализованная конструкция воздушного судна удовлетворяет требованиям по безопасности, определенным в PASA. ASA подтверждает, что требования из оценки функциональных опасностей воздушного судна и PASA выполнены. ASA также демонстрирует, что архитектура воздушного судна, а также взаимосвязи между функциями воздушного судна и системами являются приемлемыми.

ASA – своего рода итоговая оценка, охватывающая результаты и свидетельства оценок безопасности, выполненных в процессе разработки. ASA в основном включает:

- *перечень отказных состояний из FHA воздушного судна со свидетельствами того, что соответствующие требования выполнены;*
- *свидетельства того, что цели плана программы безопасности достигнуты;*
- *свидетельства того, что архитектура воздушного судна удовлетворяет качественным и количественным требованиям по безопасности;*
- *свидетельства того, что архитектура воздушного судна удовлетворяет требованиям распределения уровней гарантии разработки;*
- *статус открытых сообщений о проблемах и их последствия для воздушного судна.*

Анализ дерева отказов (FTA)

Анализ дерева отказов (Fault Tree Analysis, FTA) представляет собой нисходящий метод, позволяющий определить, какие одиночные отказы или сочетания отказов могут возникать на нижних уровнях и приводить к каждому отказному состоянию.

Основное назначение FTA – определить вероятность возникновения верхнего события и тем самым продемонстрировать выполнение требования по вероятности, заданного в документе более высокого уровня, обычно в FHA. Кроме того, FTA решает дополнительные задачи:

- *FTA позволяют оценивать предлагаемую архитектуру системы и назначать бюджеты надежности системам и изделиям;*

- *FTA выявляют необходимость модификации конструкции;*
- *повышение надежности компонентов;*
- *дополнительное резервирование;*
- *дополнительный мониторинг;*
- *увеличение объема технического обслуживания.*

Интенсивности отказов для базовых событий FTA могут браться из FMEA.

Анализ видов и последствий отказов (FMEA)

Анализ видов и последствий отказов (Failure Modes and Effects Analysis, FMEA) представляет собой систематический анализ снизу вверх. Он выполняется для выявления видов отказов системы, изделия или функции и определения воздействия отказа на следующий более высокий уровень. FMEA может выполняться на уровне компонента, функции или линейно-заменяемого блока (Line Replaceable Unit, LRU). В общем случае FMEA рассматривает индивидуальные и комбинированные последствия одиночных отказов.

Как минимум, анализ видов и последствий отказов должен включать следующее:

- *идентификацию компонента или функции;*
- *вид(ы) отказов компонента или функции;*
- *интенсивность отказов для каждого вида отказа;*
- *тяжесть последствий отказа;*
- *последствие отказа на следующем более высоком уровне;*
- *средства обнаружения отказа;*
- *компенсирующие действия (то есть автоматические или ручные).*

Аспекты безопасности ПО

В отличие от аппаратуры, где безопасность изделия можно количественно оценить через вероятность его отказа за заданный период времени, безопасность ПО нельзя измерить численно.

Поэтому для ПО не задают вероятности отказа в зависимости от тяжести отказного состояния. Вместо этого объем документирования и строгость верификации ПО устанавливают по тяжести отказного состояния, связанного с неспособностью ПО выполнять свою заданную функцию.

Категории отказных состояний ПО аналогичны категориям отказных состояний аппаратуры, описанным выше в этом обзоре. Следующая таблица взята из ED-12, эквивалентного DO-178:

Категория отказного состояния	Описание	Уровень гарантии разработки ПО
Катастрофическое	Отказные состояния, которые сделали бы невозможными продолжение безопасного полета и посадку	A
Опасное / тяжелое значительное	Отказные состояния, которые снижали бы возможности воздушного судна или способность экипажа справляться с неблагоприятными условиями эксплуатации до такой степени, что имело бы место: 1. большое снижение запасов безопасности или функциональных возможностей, 2. физические страдания или более высокая нагрузка, из-за которых на летный экипаж нельзя было бы полагаться в точном или полном выполнении своих задач, 3. неблагоприятные последствия для находящихся на борту лиц, включая серьезные травмы или потенциально смертельные травмы небольшого числа лиц на борту.	B
Значительное	Отказные состояния, которые снижали бы возможности воздушного судна или способность экипажа справляться с неблагоприятными условиями эксплуатации до такой степени, что имело бы место, например: 1. существенное снижение запасов безопасности или функциональных возможностей, 2. значительное увеличение нагрузки на экипаж или появление условий, ухудшающих эффективность экипажа, 3. дискомфорт для находящихся на борту лиц, возможно включая травмы.	C
Незначительное	Отказные состояния, которые не приводили бы к существенному снижению безопасности воздушного судна и требовали бы от экипажа действий, находящихся вполне в пределах их возможностей. Например: 1. небольшое снижение запасов безопасности или функциональных возможностей, 2. небольшое увеличение нагрузки на экипаж, например рутинные изменения плана полета, 3. некоторое неудобство для находящихся на борту лиц.	D

Анализ общих причин (ССА)

Приемлемая вероятность отказных состояний часто выводится из оценки нескольких систем при допущении, что отказы независимы. На практике такой независимости может не быть, поэтому нужны специальные исследования, подтверждающие, что независимость действительно обеспечена или что остаточная зависимость приемлема. Анализ общих причин (Common Cause Analysis, CCA) рассматривает события, которые могут привести к опасному или катастрофическому отказному состоянию. CCA делится на три области исследования:

Зональный анализ безопасности (Zonal Safety Analysis, ZSA): цель этого анализа – убедиться, что установка систем и оборудования в каждой зоне воздушного судна соответствует достаточному уровню безопасности с точки зрения конструкции и монтажа, взаимных воздействий между системами и ошибок технического обслуживания.

Анализ специфического риска (Particular Risk Analysis, PRA): специфические риски – это события или воздействия вне рассматриваемых систем, например пожар, утечки жидкостей, столкновение с птицей, электромагнитные поля высокой интенсивности (High Intensity Radiated Fields, HIRF), молния и т. д. Для каждого риска нужно выполнить отдельное исследование, чтобы проанализировать и задокументировать одновременные или каскадные последствия, которые могут нарушить независимость. Цель PRA – обеспечить, чтобы последствия, связанные с безопасностью, были устранены либо чтобы риск был приемлемым.

Анализ общего режима отказов (Common Mode Analysis, CMA): CMA выполняется для подтверждения предполагаемой независимости событий, рассматриваемых совместно для данного отказного состояния. Иначе говоря, CMA верифицирует, что сочетания событий в FTA действительно независимы в реальной реализации. Нужно анализировать последствия ошибок разработки, производства, установки, технического обслуживания и экипажа, а также отказов компонентов системы, которые нарушают эту независимость.

ПО и платформы для сертификации по ARP

Процесс сертификации, как всем известно, сложен и занимает много времени. Однако существуют стратегии и инструменты, которые улучшают процесс сертификации, уменьшают затраты ресурсов, сроки и, как следствие, стоимость сертификации. В конечном счете такие инструменты также могут повысить безопасность воздушного судна: меньше времени и ресурсов уходит на задачи, которые можно автоматизировать, а инженеры получают больше времени для проектирования и работы над безопасностью воздушного судна.

Один из инструментов, часто применяемых в критичных для безопасности проектах, – инструмент на основе моделей, создающий цифровые двойники риска (Digital Risk Twins, DRT). Платформа среды проектирования с учетом технического обслуживания MADe (Maintenance Aware Design Environment) дает дополнительные преимущества для сертификации, выявленные авторами при анализе. Эти преимущества можно разделить на краткосрочные и долгосрочные.

Краткосрочные преимущества таких инструментов:

- автоматическая генерация сертификационной документации;
- стандартизированная таксономия;
- верификация безопасности в рамках PSSA или SSA;
- помощь в заполнении материалов FHA;
- анализы на основе моделирования.

Долгосрочные преимущества таких инструментов:

- механизм накопления знаний, обеспечивающий сохранение проектных знаний внутри организации;
- повторное использование сохраненных активов знаний в других проектах;
- возможность легко адаптировать проект и повторно выпускать требуемую сертификационную документацию;
- возможность сократить разрыв до проектирования систем на основе моделей.

Итоговые замечания о безопасности воздушного судна и авионики

Процесс оценки безопасности имеет фундаментальное значение для формирования надлежащих целей безопасности воздушного судна и систем. Объем работ по оценке безопасности зависит от классификации отказных состояний на уровне воздушного судна, а также от сложности интеграции и реализации системы. Поэтому процесс оценки безопасности нужно планировать на раннем этапе разработки и управлять им на протяжении всей разработки.

Если в ходе PSSA при оценке архитектуры системы FTA показывает, что система не удовлетворяет требованиям по безопасности, то есть рассчитанная вероятность нежелательного верхнего события превышает допустимую, производитель системы может выполнить несколько ключевых улучшений:

- использовать компоненты с меньшей интенсивностью отказов;
- улучшить встроенный контроль (Built-In Test, BIT), чтобы обнаруживать большую долю отказов;
- увеличить степень резервирования в системе.

Что из перечисленного лучше? Ответ зависит от стоимости, доступности и простоты реализации; чтобы выбрать наилучшую альтернативу для конкретной системы, нужен анализ компромиссов.

Теперь вы начали оценки безопасности воздушного судна и систем авионики с использованием ARP4761A и готовы перейти к разработке собственно воздушного судна и систем авионики. Добро пожаловать в ARP4754A (пора перевернуть страницу...).

ARP4754A: разработка воздушных судов и систем

Автор: Вэнс Хилдерман

Рецензенты:

- г-н Кевин Крозье, старший инженер по ПО авионики, консультант – представитель Федерального управления гражданской авиации США (Federal Aviation Administration, FAA) по инженерной экспертизе (Designated Engineering Representative, DER), сертифицированный FAA летный инструктор и коммерческий пилот.
- г-н Рассел Макмаллан (бывший технический директор компании BECA)

Еще немного не магических чисел: 4754, 4761, 79...

Как показано в ARP4761A применительно к оценкам безопасности, безопасность опирается не на магические числа, а на практические ответы. Точно так же безопасность никогда не возникает случайно, но настоящая безопасность должна предотвращать происшествия. Числа “4754”, “4761” и “79” не являются магическими, однако связаны с безопасностью. Подходы к безопасности и связанные с ними документы эволюционировали: новые ответы содержатся в редакциях 4754A и 4761A, в частности в документе SAE ARP4754A (европейское обозначение ED-79) и в ARP4761A.

Старые времена

Теперь вы знаете, что “старые времена” в технологиях – это любое время до прошлого года. В очень старые времена (высадки на Луну, космические шаттлы, коммерческие самолеты с четырьмя двигателями) безопасность обеспечивали умом и доводкой: хорошие инженеры делали все возможное, чтобы предотвратить происшествия, а затем конструкцию дорабатывали там, где даже лучших усилий оказывалось недостаточно. В очень старые времена вычислительные ресурсы были скромнее, графики – свободнее, и обычно удавалось получать приемлемый уровень безопасности: у космических шаттлов вероятность не взорваться составляла 98,5 процента, а катастрофы коммерческих самолетов со смертельным исходом происходили “всего лишь” несколько раз в год. Затем очень старые времена уступили место просто старым временам, и очень старые методы стали работать хуже. Одни говорили, что инженеры стали не такими умными, другие – что эти инженеры пытались сделать компьютеры слишком умными. Но все соглашались в одном: нужен более формализованный подход к безопасности, а SAE уже десятилетиями занималась документами серии рекомендуемых аэрокосмических практик (Aerospace Recommended Practices, ARP).

SAE ARP4754, *“Соображения по сертификации высокоинтегрированных или сложных авиационных систем” (Certification Considerations for Highly-Integrated or Complex Aircraft Systems)*, был опубликован в ноябре 1996 года. Тесно связанный с ним документ ARP4761, *“Руководящие указания и методы выполнения процесса оценки безопасности гражданских бортовых систем и оборудования” (Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment)*, вышел в следующем месяце, в декабре 1996 года.

ARP4754A сегодня

Официальное название пересмотренного ARP4754A – *“Руководящие указания по разработке гражданских воздушных судов и систем” (Guidelines for Development of Civil Aircraft and Systems)*. Документ охватывает цикл разработки воздушных судов и авиационных систем. Обычно о книге нельзя судить по обложке или названию, но в этом случае название действительно несет важный сигнал: если вы участвуете в разработке воздушных судов или систем, руководящие указания ARP4754A нужно знать хорошо. Почему? Перед первым открытием ARP4754A стоит понять два ключевых момента:

1. В названии ARP4754A сказано “руководящие указания”, но непонимание и неприменение ARP4754A может снизить безопасность и существенно уменьшить шансы добиться сертификации. Возможность продемонстрировать надежную и безопасную авионику начинается с подхода к безопасности системы еще до разработки. Задним числом применить такой подход, чтобы исправить слабую систему, крайне трудно.
2. Хотя предшественник, ARP4754, был в целом похож, слишком многие организации трактовали его как “необязательный” документ, будто само слово “guideline” это оправдывало. Однако сертифицирующие органы по всему миру все чаще и уже формально требуют соблюдения именно последней редакции, ARP4754A.

Для опытных и компетентных разработчиков воздушных судов или авиационных систем ARP4754A читается как книга о поддержании здоровья: составьте план, поймите принципы здорового образа жизни, соблюдайте меры безопасности, правильно питайтесь, снижайте стресс, занимайтесь физическими упражнениями, высыпайтесь, проходите регулярные обследования, чтобы подтвердить соблюдение плана, и повторяйте этот цикл. Для воздушного судна аналогичное краткое изложение ARP4754A выглядело бы так:

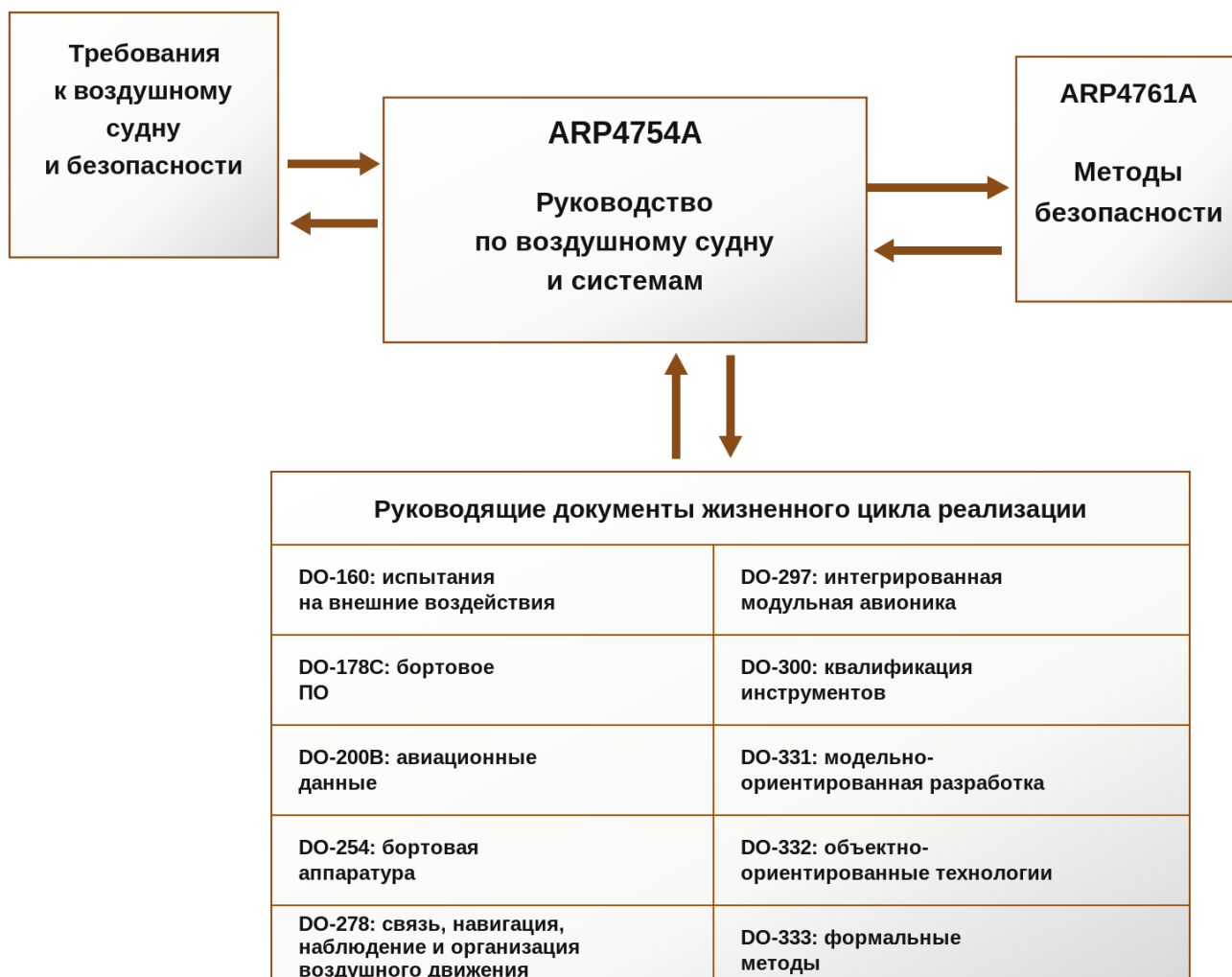
1. Спланируйте экосистему жизненного цикла разработки воздушного судна/системы;
2. Выполняйте мероприятия по безопасности согласно ARP4761 (с 2018 года – ARP4761A);
3. Определите и обоснуйте уровень гарантии разработки;
4. Определите архитектуру системы и требования; выполните валидацию;
5. Выполняйте верификацию и управление конфигурацией;

6. Обеспечьте гарантию процесса и подтвердите критерии перехода.

Помните об экосистеме безопасности, систем, аппаратуры и ПО, показанной ниже:



Как в переносном, так и в буквальном смысле разработка систем по ARP4754A занимает центральное место: ей предшествует оценка безопасности, которую необходимо учитывать при определении архитектуры воздушного судна/системы и требований безопасности к ним. В свою очередь, разработка систем предшествует разработке ПО и аппаратуры, но системные аспекты продолжают рассматриваться на протяжении всей разработки ПО и аппаратуры. Уточненный взгляд на соответствующие руководящие документы показан здесь:



Зачем нужен ARP4754A? Предпосылки

Прежде чем переходить к деталям ARP4754A, стоит всерьез задуматься, зачем он вообще существует. Несколько десятилетий назад, когда авиационные системы были проще, грамотные проектировщики могли мысленно охватить такие системы и сразу переходить к реализации. Нужно признать, что и сегодня потребность в ARP4754A менее очевидна для простых систем. В то же время количество, разнообразие и сложность систем продолжают расти экспоненциально. Очевидно, что авиационные системы могут быть намного сложнее обычных капитальных коммерческих зданий, но было бы немыслимо начинать строительство офисного здания без анализа грунта и сейсмики, проекта фундамента и плана инспекций. Эти инспекции, разумеется, продолжаются на протяжении всего строительства и охватывают электрику, водоснабжение, аварийные выходы и надлежащее усиление конструкции. Выдающиеся строители, возможно, и смогли бы возвести безопасное здание без подробных планов, чертежей, процессов и инспекций, но полностью подтвердить его “величие” было бы невозможно. Почему? “Величие” должно подтверждаться доказательствами. Доказательство строится на оценке реализации относительно планов и последующем исправлении найденных недостатков.

Очевидно, что без планов и процессов для здания невозможно оценивать или утверждать, что его проект и строительство безопасны. Поскольку разработка авионики может быть сложнее строительства здания, авиационным системам требуются общее планирование, процессы, требования, безопасная разработка, верификация и валидация, а также доказательства соответствия. Добро пожаловать в ARP4754A, *“Руководящие указания по разработке гражданских воздушных судов и систем”*.

Предпосылки: ARP4754

Первоначальный стандарт ARP4754 был опубликован в 1996 году, чтобы помочь организациям, разрабатывающим авионику, мыслить шире, чем только аппаратура и ПО. Напомним: DO-178 (и его европейский эквивалент ED-12) был опубликован более чем за десять лет до этого и давал руководящие указания по авиационному ПО. Но к началу 1990-х стало ясно, что безопасное ПО и сама сертификация ПО требуют как понимания системы, так и подтверждения аспектов безопасности на уровне системы. ARP4754 был ориентирован на авиационные системы, отказ которых потенциально мог повлиять на безопасность воздушного судна или людей на борту. Хотя на воздушном судне, безусловно, есть критичные автономные компоненты, которые могут влиять на безопасность, ARP4754 сосредоточен не на компонентах, а на системах со сложными взаимодействиями с другими системами на борту воздушного судна или вне его. Такие системы обычно охватывают несколько областей знаний и, вероятно, будут со временем развиваться.

Поэтому их разрабатывают разные люди, в разных дисциплинах, часто разделенные местом и временем. Лучший способ обеспечить безопасную реализацию – кодифицированные процессы разработки, основанные на детерминированном подходе к безопасности: ARP4754.

Первоначальный стандарт ARP4754 был также написан до появления более поздних тенденций в разработке авионики, таких как разработка на основе моделей (Model-Based Development, MBD) и интегрированная модульная авионика (Integrated Modular Avionics, IMA). Его акцент был сделан на нисходящем итерационном подходе к разработке авиационных систем: функциональность отдельных систем выявляется, а затем все более подробно уточняется. На момент появления стандарта процесс обеспечения безопасности воздушного судна и систем был менее зрелым, а ARP4761 существовал лишь как концепция. Кроме того, бытовало представление, что ARP4754 является документом SAE и потому не является настоящим специализированным руководством по авионике. Нужно признать и то, что сертифицирующие органы обычно не всегда требовали доказательств его соблюдения. ARP4754 был хорошим справочным материалом по разработке систем, но перечисленные выше обстоятельства не позволили ему стать выдающимся документом. Отсюда и возникла необходимость в ARP4754A...

ARP4754A по сравнению с ARP4754

ARP4754 в исходной форме, до его обновления до ARP4754A, обладал следующими характеристиками:

Старое название: *“Соображения по сертификации высокоинтегрированных или сложных авиационных систем”*

Новое название: *“Руководящие указания по разработке гражданских воздушных судов и систем”*

Как легко видеть, новое название ARP4754A:

- подчеркивает аспекты разработки;
- устраняет неоднозначность терминов “высокоинтегрированные” и “сложные”.

Серьезно: просто спросите, какие системы на современных воздушных судах не являются ни сложными, ни высокоинтегрированными. Ответ: “Их все меньше и меньше...”. Разработчику системы может казаться, что его система не сложна, потому что все инженеры проекта легко ее понимают. Кроме того, каждая система физически и электрически отделена от других систем, поэтому ее можно считать “не высокоинтегрированной”. Однако большинство авиационных систем на самом деле интегрированы: при их оценке безопасности, проектировании, установке и эксплуатации нужно учитывать воздушное судно и другие его системы. Отсюда и пересмотренное, менее субъективное название ARP4754A.

С редакцией “А”, то есть ARP4754A, было внесено несколько ключевых улучшений, как показано ниже:

Ключевые атрибуты и обоснование рекомендуемой аэрокосмической практики ARP4754A (Aerospace Recommended Practice, ARP)

- Более тесная увязка с рекомендуемой аэрокосмической практикой ARP4761A по безопасности, DO-178C по ПО и DO-254 по бортовой электронной аппаратуре.
- Повышенное внимание к аспектам разработки воздушного судна и гарантии.
- Термин «гарантия проектирования» заменен термином «уровень гарантии разработки элемента».
- Больше ясности, меньше неоднозначности и согласованная терминология.
- Фактически обязательное требование использовать и применять рекомендуемую аэрокосмическую практику ARP4754A.

ARP4754 был “хорошим”: он описывал базовый процесс разработки безопасных и качественных авиационных систем и воздушных судов. Однако из-за эволюции связанных руководящих документов, уточнения сертификационных подходов и необходимости учитывать растущую интеграцию и сложность систем многие считали ARP4754

неполным, поэтому он не применялся с требуемой строгостью. “Итерационный подход” ARP4754 был расширен в ARP4754A (совместно с ARP4761), чтобы разработчики применяли формальный итерационный процесс безопасности, при котором безопасность непрерывно рассматривается с помощью уточняемых оценок безопасности на протяжении всего проекта.

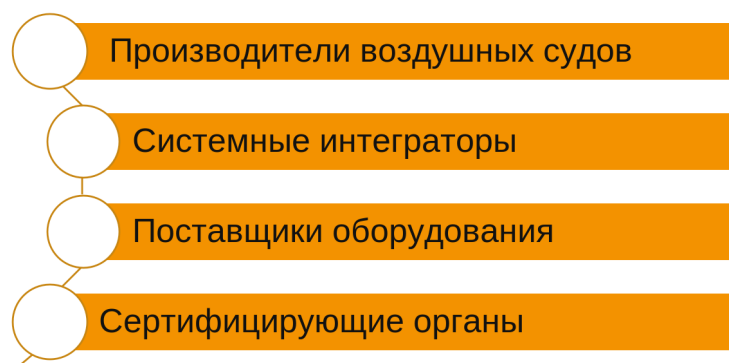
В отличие от него, ARP4754A действительно подчеркивает важность всей экосистемы разработки авиационных систем, основанной на формальном процессе безопасности (при поддержке ARP4761). ARP4754A дает конкретные указания по соблюдению нормативных требований и превращает их в практическое руководство по разработке воздушных судов и систем, подчеркивая необходимость непрерывной интеграции процесса безопасности и системного процесса на всем протяжении разработки.

В одном недавнем проекте по авионике автор проводил аудит документов разработчика системы, которые в авиационной сертификационной среде обычно называют “артефактами”. Процедуры клиента больше походили на планы, а его аудиты – на рассмотрения. Очевидно, настало время вернуться к основам. Хотя на эту тему легко написать целые главы, гораздо труднее выразить все одной фразой. Писателю Марку Твену приписывают фразу, примерно соответствующую первой строке десятистраничного письма:

“Простите за длину этого письма; у меня не было времени сделать его короче”. Смысл в том, что краткие резюме даются труднее, чем целые романы. В авионике длинные романы не любят: в них слишком много лишнего, а у серьезных разработчиков нет на это времени. Вот предельно краткое резюме планов, процессов, рассмотрений и аудитов для авиационных систем:

- “Планы” учитывают требования безопасности и кратко излагают, что именно вы будете делать;
- “Процессы” описывают, как вы будете выполнять планы;
- “Контрольные перечни” задают объективные критерии рассмотрения, позволяющие определить, соблюдались ли процессы;
- “Аудиты гарантии процесса” оценивают соответствие инженерной и производственной деятельности, включая соответствующие переходы к этим процессам и рассмотрениям.

Кто же является предполагаемыми пользователями ARP4754A? Основные заинтересованные стороны ARP4754A кратко показаны на следующем рисунке:



Основные заинтересованные стороны ARP4754A

Помните притчу о разных слепых людях, ощупывающих слона? Так же многие новички ARP4754A видят документ по-разному в зависимости от собственной перспективы: руководство по безопасности и проектированию воздушного судна, руководство по разработке систем или практическое руководство по интеграции режимов оценки безопасности ARP4761A. Так что же это на самом деле? ARP4754A – все это сразу, но в формате общего руководства по поддержанию формы, основанного на доказуемой науке. Это показано на рисунке ниже:

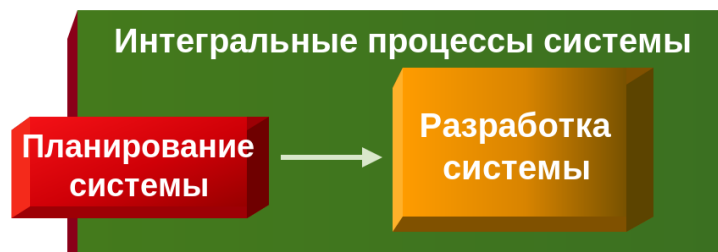


Основа ARP4754A

Три ключевых процесса ARP4754A

В предельно упрощенном виде ARP4754A требует выполнить процесс системного планирования, охватывающий восемь тем планирования, описанных ниже. После того как эти восемь тем планирования ARP4754A отражены в утвержденных планах, начинается процесс системной разработки. Системная разработка в рамках ARP4754A не включает собственно разработку аппаратуры или ПО, поскольку они рассматриваются соответственно в DO-254 для бортовой аппаратуры и DO-178C для бортового ПО. На фоне этого непрерывно выполняются интегральные процессы системы. Связь между этими процессами показана на рисунке ниже. Обратите внимание: стрелка между

системным планированием и системной разработкой намеренно проведена в одну сторону, а не в обе. После утверждения планов заранее не предполагается, что эти уже утвержденные планы будут обновляться. Если позднее появляются отклонения от планов, эти отклонения утверждаются и документируются вне самих планов.



Процесс планирования по ARP4754A

Что общего у воздушных судов, сложных систем и простых систем? Все они подпадают под область применения ARP4754A. Почти невозможно, чтобы одно руководство подходило всем для всего, но ARP4754A пытается к этому приблизиться. Поскольку воздушные суда и системы могут очень сильно различаться, ARP4754A не предписывает жестко регламентированный подход. Вместо этого разработчику предлагается рассмотреть последствия для безопасности и функциональности, как внешние, так и внутренние по отношению к области его разработки, и действовать соответственно.

Аналогия здесь – персональный тренер по физической подготовке, цели которого резко различаются в зависимости от того, тренирует ли он травмированного пациента, обычного офисного работника или олимпийского спортсмена: разные предметные области требуют разного фокуса. Но ARP4754A один, поэтому его руководящие указания должны быть достаточно общими, чтобы удовлетворять разные области. Центральное место в любом процессе разработки воздушного судна/системы занимает требуемый ARP4754A процесс планирования.

ARP4754A дает комплексные руководящие указания по выполнению множества инженерных мероприятий, необходимых для разработки безопасных и качественных авиационных систем. Как знают все пилоты, перед значительным полетом обычно полезно подать план полета до взлета. Такой план полета выполняет три основные цели:

Три основные цели плана полета

1. Заставить пилота рассмотреть и указать фактический маршрут с учетом рельефа, погоды, ограничений воздушного пространства и других факторов.
2. Предупредить органы управления о том, где и когда пилот намерен выполнять полет.
3. Дать средство проверки запланированных пилотом местоположений при отклонениях.

Таким образом, процесс планирования ARP4754A аналогичен плану полета, поскольку у него есть три основные цели:

Три основные цели процесса планирования рекомендуемой аэрокосмической практики ARP4754A (Aerospace Recommended Practice, ARP)

1. Заставить разработчиков систем авионики учитывать и задавать ключевые инженерные процессы с учетом требуемых последствий для безопасности.
2. Информировать сертификационные органы о технических характеристиках и характеристиках безопасности планируемой системы, а также о соответствующих запланированных инженерных работах.
3. Дать средство проверки плановых и фактических инженерных работ по авионике, обеспечивая ответственность разработчиков.

Могут ли планы разработки системы изменяться после их определения? Разумеется. Так же как планы полета могут меняться, реальная жизнь разработки авионики полна небольших изменений. Однако изменения в процессе планирования по ARP4754A должны тщательно учитывать общие аспекты безопасности, как внутри этой системы, так и по отношению к другим системам – воздушным и наземным. И они должны быть документированы и утверждены. Процесс планирования ARP4754A требует рассмотреть и документировать следующие ключевые аспекты разработки, показанные ниже:

Восемь тем планирования рекомендуемой аэрокосмической практики ARP4754A (Aerospace Recommended Practice, ARP)

1. **Разработка.** Определить процессы и методы разработки архитектуры, интеграции и реализации.
2. **Безопасность.** Определить область работ по безопасности, применимую к воздушному судну или системе.
3. **Управление требованиями.** Определить сбор требований и управление ими.
4. **Валидация.** Определить методы подтверждения корректности требований и допущений.
5. **Верификация реализации.** Определить процессы и критерии оценки соответствия реализации требованиям.
6. **Управление конфигурацией.** Определить управление конфигурационными единицами и версиями в жизненном цикле.
7. **Гарантия процесса.** Определить независимые действия, подтверждающие соблюдение процессов и планов разработки.
8. **Сертификация.** Определить порядок достижения сертификации.

То, как перечисленные выше детали планирования распределяются по документам, менее важно, чем сами детали планирования. Обычно по каждому из этих аспектов составляют отдельные планы (хотя верификацию и валидацию часто объединяют); отдельные плановые документы повышают возможность их повторного использования в будущих проектах. Всегда возникает вопрос: “Насколько подробными должны быть планы?” Ответ, разумеется, зависит от сложности соответствующих инженерных мероприятий. Планирование строительства небоскреба требует больше деталей, чем планирование небольшого дома. То же самое относится к воздушным судам и системам. Планы должны быть более подробными в тех областях системы, которые определены как рискованные, но при этом нужно признать, что вы можете не знать о риске, пока не начнете работу. В приложении А к ARP изложены объективные требования к процессам: независимость требуется, рекомендуется, согласуется по договоренности или не требуется. Детализации должно быть достаточно, чтобы детерминированно ответить на два вопроса:

1. Может ли сертифицирующий орган рассмотреть планы и определить, что описанные мероприятия способны привести к доказуемо безопасной разработке?
2. Можно ли составить подробные контрольные перечни для каждого мероприятия планирования, извлекая информацию из планов для независимой оценки соответствующих мероприятий разработки?

Область охвата процесса планирования ARP4754A

Элемент планирования	Описание элемента	Раздел ARP4754A
Разработка	Установить процесс и методы, которые будут использоваться для формирования основы разработки архитектуры воздушного судна/системы, интеграции и реализации.	Этот раздел и 4.0
Программа безопасности	Установить область и содержание мероприятий по безопасности, связанных с разработкой воздушного судна или системы.	5.1.5
Управление требованиями	Определить и описать, каким образом требования фиксируются и управляются. Иногда эти элементы включаются совместно с элементами валидации.	5.3
Валидация	Описать, как будет показано, что требования и допущения являются полными и корректными.	5.4
Верификация реализации	Определить процессы и критерии, применяемые при демонстрации того, что реализация удовлетворяет предъявленным к ней требованиям.	5.5
Управление конфигурацией	Описать ключевые единицы конфигурации, связанные с разработкой, и порядок управления ими.	5.6
Гарантия процесса	Описать средства гарантии того, что практики и процедуры, применяемые при разработке системы, соблюдаются.	5.7
Сертификация	Описать процесс и методы, которые будут использоваться для получения сертификации.	5.8

Краткое замечание о валидации, верификации, гарантии процесса, рассмотрении и аудитах

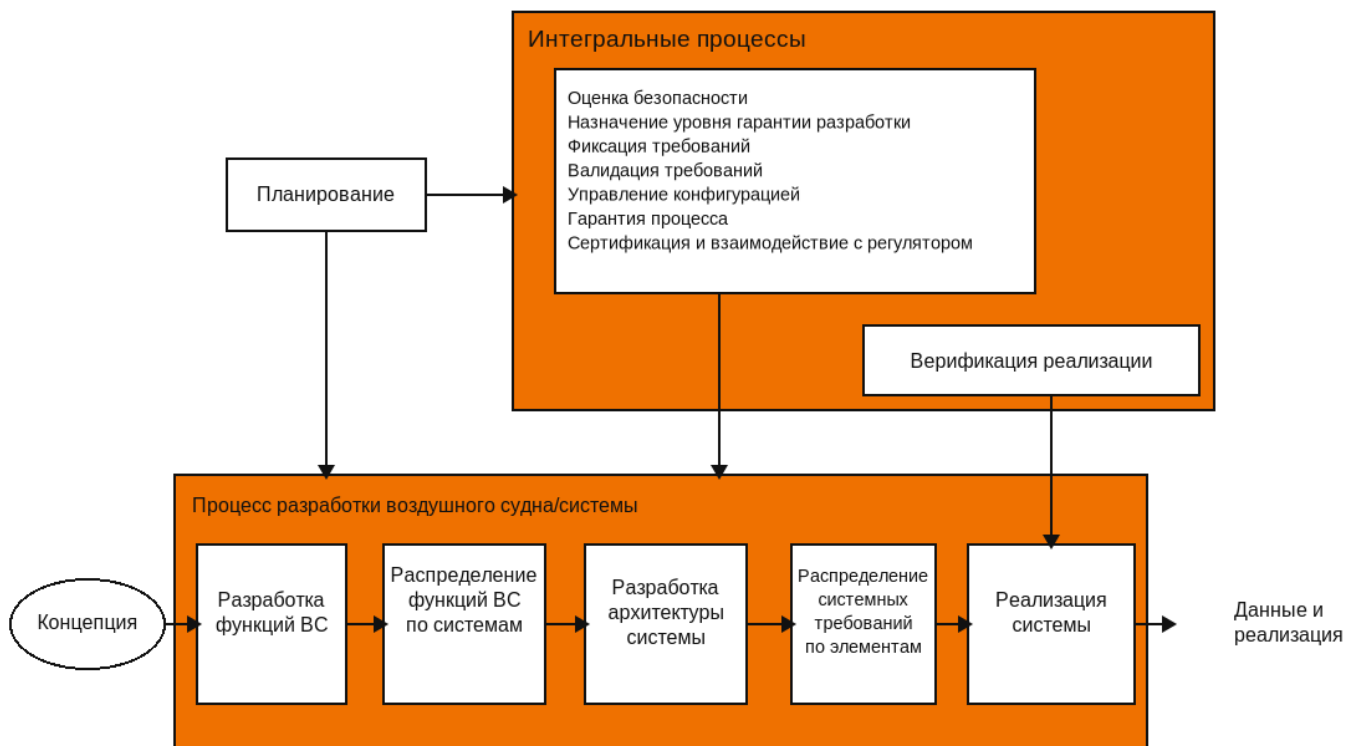
Верификация выполняется инженерным персоналом и оценивает, соответствует ли реализация требованиям. Валидация оценивает приемлемость самих требований. В мире авионики (а если читать между строк ARP4754A и документов серии DO, то особенно) стоит понимать, что в самом простом виде “верификация = рассмотрение + испытания + анализ”. Рассматривается практически все. Требования и реализация испытываются. Анализ применяется тогда, когда рассмотрение и испытания не дают полностью определенного вывода. И валидация, и верификация рассматриваются далее.

Аудиты ориентированы на процесс и выполняются независимым персоналом гарантии качества (Quality Assurance, QA), который также называют персоналом гарантии процесса (Process Assurance, PA). Это не рассмотрение: аудиты определяют, следовали ли инженеры установленным процессам. Для систем гарантия качества именуется гарантией процесса. Она является надмножеством гарантии качества, но в авиации эти термины часто используют почти взаимозаменяемо.

Серьезная проблема возникает, когда рассмотрение неполны, потому что инженер думает, будто позже все посмотрит аудитор. Не посмотрит. Аудиторы не выполняют рассмотрение, точка. Они выполняют аудиты. Рассмотрения технические, аудиты процессные. Нужны и те и другие: каждое важно, но это разные вещи. Как бутылка вина и само вино: невозможно наслаждаться одним без другого. Что важнее? Оба.

Еще одна проблема возникает, когда аудиторы считают себя рецензентами. Это не так. Если аудитор (гарантия процесса) вынужден выполнять рассмотрение, возникает вопрос: почему инженерный персонал не выполнил надлежащее рассмотрение, документированное соответствующим контрольным перечнем?

В целом процесс ARP4754A показан ниже:



Общий процесс ARP4754A

Безопасность прежде всего

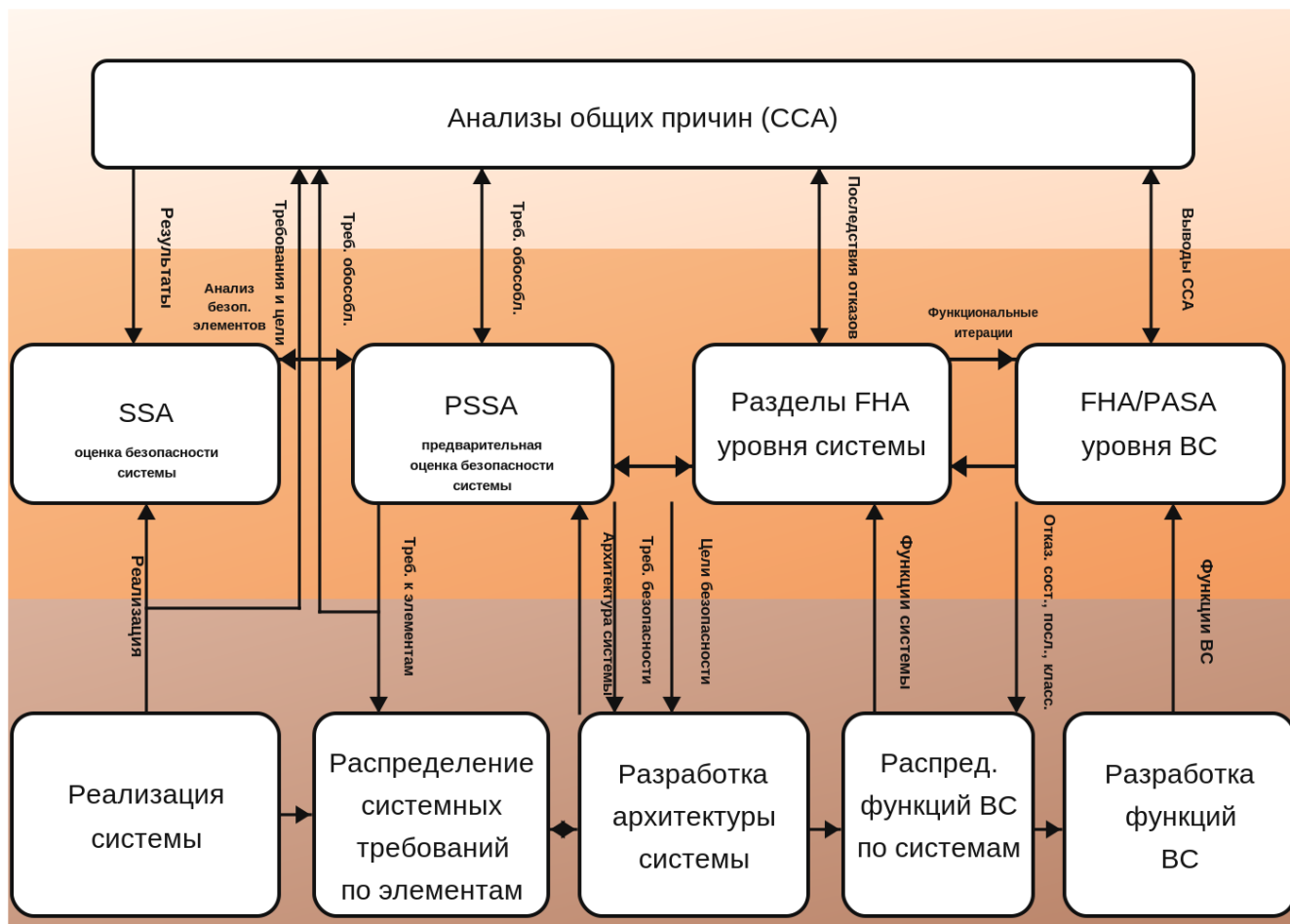
Вы спланировали разработку своей авиационной системы в соответствии с изложенным выше, и планы завершены. С чего начать? Просто: с процесса безопасности. Безопасность действительно является основой и главным фокусом ARP4754A. Все эти планы, процессы и контрольные перечни нужны для того, чтобы безопасность оставалась в центре разработки и оценивалась на всем ее протяжении.

На уровне воздушного судна ARP4754A требует общий план программы безопасности (Safety Program Plan, SPP). Обычно для каждой системы составляют планы программы безопасности на уровне системы. В любом случае каждый аспект рассматривается в рамках плана программы безопасности уровня воздушного судна. Как и при строительстве дома, область безопасности авионики принимает разные формы: фундамент, архитектура, обнаружение проблем, снижение риска проблем, восприимчивость к конкретным опасным воздействиям (например, молнии) и верифицируемость. Поэтому требуются несколько мероприятий и анализов по безопасности. Следующий рисунок, обобщающий основные мероприятия по безопасности, повторен ниже:

Основные мероприятия по безопасности авионики

- 1. Оценка функциональных опасностей (Functional Hazard Assessment, FHA).**
Выявить потенциальные отказы и их последствия, затем классифицировать тяжесть каждого отказа.
- 2. Предварительная оценка безопасности воздушного судна (Preliminary Aircraft Safety Assessment, PASA) или предварительная оценка безопасности системы (Preliminary System Safety Assessment, PSSA).** Проанализировать предлагаемую архитектуру, определить, как могут возникать отказы, выявленные при оценке функциональных опасностей, и сформировать требования безопасности.
- 3. Оценка безопасности воздушного судна (Aircraft Safety Assessment, ASA) или оценка безопасности системы (System Safety Assessment, SSA).** Оценить системы воздушного судна, чтобы определить, выполнены ли требования по безопасности.
- 4. Анализ общих причин (Common Cause Analysis, CCA).** Проверить, достаточна ли независимость функций и систем для заданной безопасности.

Более полное описание оценки функциональных опасностей (Functional Hazard Assessment, FHA), предварительной оценки безопасности системы (Preliminary System Safety Assessment, PSSA), оценки безопасности системы (System Safety Assessment, SSA) и анализа общих причин (Common Cause Analysis, CCA) см. в соответствующей главе этой книги, посвященной оценке безопасности по ARP4761. Краткое изложение этих процессов показано ниже:



Процесс оценки безопасности

Оценка функциональных опасностей (ФНА)

Цель оценки функциональных опасностей состоит в том, чтобы выявить функции, отказные состояния функций (потеря функции, неправильная работа и т. п.) и их классификацию (катастрофическое, опасное и т. п.), чтобы можно было предложить и реализовать такие конструкции воздушного судна и системы, которые снижают вероятность возникновения отказных состояний до приемлемо низких уровней. Нужно отметить, что в некоторых случаях полная потеря функции влияет меньше, чем частичная потеря функциональности или вводящая в заблуждение функциональность: пилот с большей вероятностью сможет компенсировать полную потерю функциональности одной системы, переключившись на другую систему.

Предварительная оценка безопасности системы (PSSA)

Предварительная оценка безопасности системы представляет собой набор анализов, обычно выполняемых на этапах требований к системе и требований к элементам в жизненном цикле воздушного судна. Именно в ходе предварительной оценки безопасности системы оцениваются и определяются предлагаемые архитектуры воздушного судна и системы; это позволяет вывести требования безопасности к системе и

элементам. Входными документами для предварительной оценки безопасности системы являются оценка функциональных опасностей воздушного судна, предварительный анализ дерева отказов (Fault Tree Analysis, FTA) воздушного судна с распределением бюджета отказов относительно верхнеуровневого требования по отказу, а также оценки функциональных опасностей систем. Документы, создаваемые в ходе предварительной оценки безопасности системы, включают обновленные оценки функциональных опасностей воздушного судна и систем, обновленные анализы дерева отказов воздушного судна и систем, а также предварительные анализы общих причин для системы. Предварительная оценка безопасности системы является непрерывным и итерационным процессом. Ее цель – определить, каким образом отказы воздушного судна и системы могут приводить к опасным состояниям, выявленным при оценке функциональных опасностей, и определить, как могут быть выполнены соответствующие требования. Предварительную оценку безопасности системы часто связывают с предварительной оценкой безопасности воздушного судна. Важное замечание: приложение 1 к консультативному циркуляру АС 23.1309-Е содержит полезные указания, помогающие оценивать серьезность отказных состояний функций и систем воздушного судна на этом этапе проекта.

Оценка безопасности системы (SSA)

Оценка безопасности системы по ARP4754A / ARP4761A верифицирует, что реализованные конструкции воздушного судна и систем соответствуют требованиям оценки функциональных опасностей воздушного судна и системы, а также предварительной оценке безопасности воздушного судна (Preliminary Aircraft Safety Assessment, PASA) и предварительной оценке безопасности системы. Документация оценки безопасности системы, формируемая в ходе этой оценки, включает:

- *обновленные анализы дерева отказов воздушного судна, оценки функциональных опасностей систем и анализы дерева отказов систем;*
- *документацию, показывающую требования к установке элементов;*
- *любые материалы, используемые для валидации классификаций отказных состояний;*
- *при необходимости – пересмотренные руководства по техническому обслуживанию, подробно описывающие новые задачи технического обслуживания, направленные на сокращение времени воздействия на компоненты;*
- *при необходимости – пересмотренные руководства по эксплуатации для летного экипажа, подробно описывающие процедуры, которым следует следовать при определенных отказных состояниях.*

Оценка безопасности системы должна верифицировать, что все существенные эффекты, выявленные в сводке видов и последствий отказов (Failure Modes and Effect Summary, FMES), рассмотрены на предмет включения в качестве первичных событий в анализ дерева отказов. Она также должна включать применимые результаты анализа общих причин.

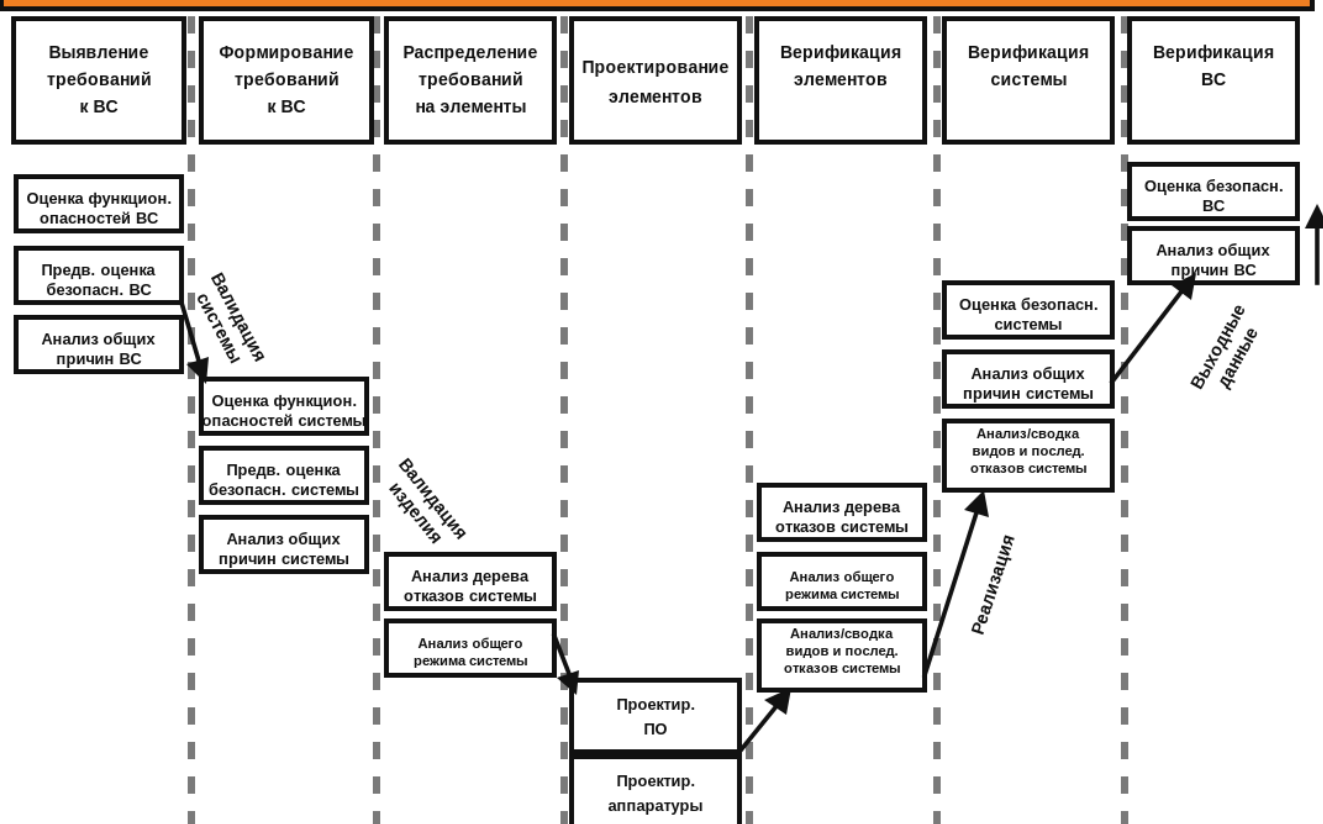
Анализ общих причин (ССА)

Принятие достаточной вероятности отказных состояний часто выводится из оценки нескольких систем на основе предположения о независимости отказов. На практике такой независимости может не быть, поэтому требуются специальные исследования, чтобы независимость можно было либо гарантировать, либо признать приемлемой. Анализ общих причин касается событий, которые могут привести к опасному или катастрофическому отказному состоянию. Одинаковое ПО, установленное на нескольких линейно-заменяемых блоках (Line Replaceable Unit, LRU), многие считают наиболее часто упускаемой областью общей причины. Например, дублированные системы управления полетом (Flight Management Systems, FMS) могут отказать в точно одинаковых условиях, если на обоих блоках установлено одно и то же ПО.

Более полное описание оценки функциональных опасностей, предварительной оценки безопасности системы, оценки безопасности системы и анализа общих причин см. в соответствующей главе этой книги, посвященной оценке безопасности по ARP4761.

Ниже показана упрощенная связь мероприятий оценки безопасности с процессами разработки по ARP4754A:

Процесс ARP4754A



Назначение уровня гарантии разработки (DAL)

Совместно с перечисленными выше мероприятиями по безопасности определяется уровень гарантии разработки (Development Assurance Level, DAL). Неформально называемый “уровнем критичности”, уровень гарантии разработки представляет собой обозначение, которое определяет строгость процесса разработки, связанную с соответствующей вероятностью возникновения ошибок. Чем более критичен элемент для безопасности полета, тем ниже должна быть вероятность возникновения ошибок в этом элементе.

ARP4754A определяет два типа уровня гарантии разработки:

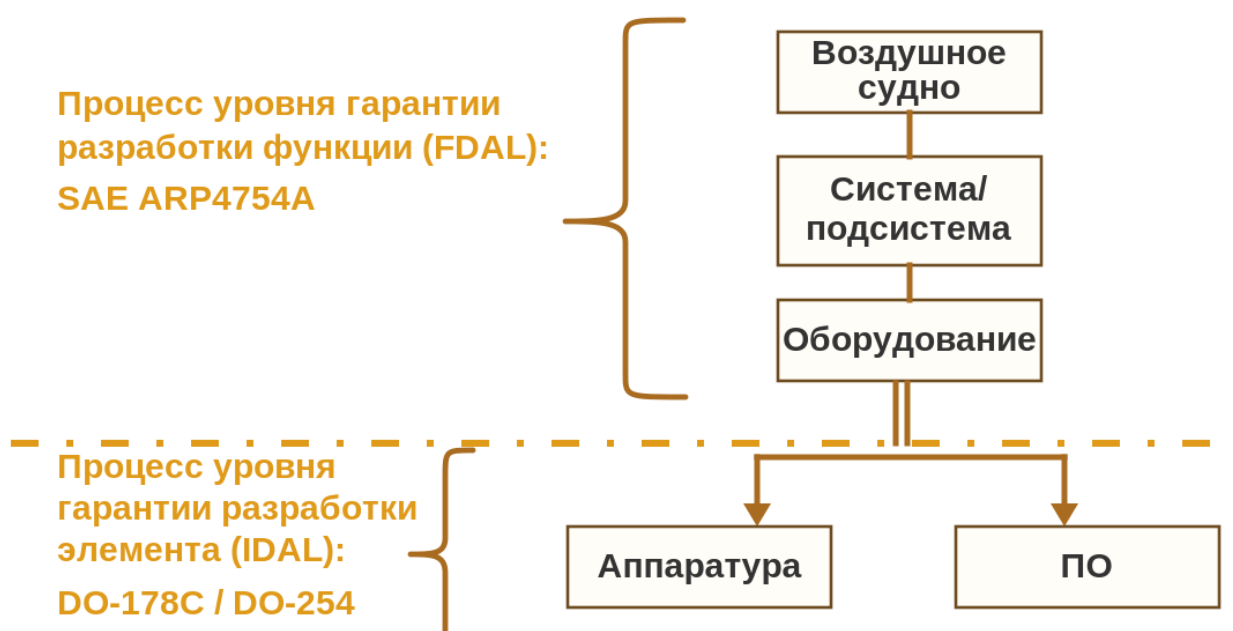
1. Уровень гарантии разработки функции (Function Development Assurance Level, FDAL);
2. Уровень гарантии разработки элемента (Item Development Assurance Level, IDAL).

Уровень гарантии разработки функции и уровень гарантии разработки элемента обобщены на рисунке ниже:

Сводка: уровень гарантии разработки функции и уровень гарантии разработки элемента

- **Уровень гарантии разработки функции (Function Development Assurance Level, FDAL).** Уровень строгости мероприятий по гарантии разработки, выполняемых для функций.
- **Уровень гарантии разработки элемента (Item Development Assurance Level, IDAL).** Уровень строгости мероприятий по гарантии разработки, выполняемых для элементов.

Следующий рисунок показывает разные варианты распределения между уровнем гарантии разработки функции и уровнем гарантии разработки элемента:



Уровень гарантии разработки функции и уровень гарантии разработки элемента: перспектива воздушного судна, системы и элемента

Помните: процесс обеспечения безопасности является итерационным, однако начало назначения уровня гарантии разработки функции должно предшествовать назначению уровня гарантии разработки элемента. Если вы не знаете уровень гарантии разработки функции, вы не можете полностью назначить последующие цели для элемента в соответствии с этим уровнем:

Два типа уровня гарантии разработки: для функции и для элемента

- **Уровень гарантии разработки функции (Function Development Assurance Level, FDAL).** Применяется к функциональным требованиям.
- **Уровень гарантии разработки элемента (Item Development Assurance Level, IDAL).** Применяется к разработке элемента.

Хотя это не обязательно, чаще всего сначала определяют уровень гарантии разработки функции, а затем уровень гарантии разработки элемента. Почему? Функции связаны с требованиями, а эти требования распределяются по элементам. Следовательно, функции и их требования описывают, “что” делает система, тогда как само требование реализуется в аппаратных и/или программных элементах. Это позволяет разделить одну функцию между несколькими элементами, что потенциально дает возможность назначить более низкий уровень гарантии разработки для отдельных элементов. Уровень строгости разработки функциональных требований определяется уровнем гарантии разработки функции, где уровень А является самым строгим. Требуемые цели разработки для каждого уровня гарантии разработки функции приведены в приложении А самого ARP4754А. Соответствующая разработка аппаратных и программных элементов также имеет специальное назначение уровня гарантии разработки элемента; требуемые цели разработки аппаратуры и ПО для каждого уровня гарантии разработки элемента приведены в DO-178С (европейское обозначение ED-12С) и DO-254 (европейское обозначение ED-80).

Таблица назначения уровня гарантии разработки функции приведена ниже:

Классификация серьезности верхнеуровневого отказного состояния:	Соответствующее назначение уровня гарантии разработки функции
Катастрофическое	A
Опасное/тяжелое	B
Значительное	C
Незначительное	D
Без влияния на безопасность	E

Варианты назначения уровня гарантии разработки функции (FDAL)

ARP4754А описывает специальные процессы учета независимости между системами и процессами в той мере, в какой они относятся к уровню гарантии разработки. Цель состоит в том, чтобы обеспечить требуемые уровни необходимой безопасности и исключить отказы общего режима. Например, если единичный отказ может затронуть несколько систем или компонентов, можно добавить независимость, чтобы этот единичный отказ не затрагивал более одной системы или компонента.

Функциональная независимость обеспечивает, чтобы функции не подвергались влиянию общих ошибок внутри одной и той же функции. Например, навигация, очевидно, является важной функцией для безопасности воздушного судна; использование двух источников навигационных данных – глобальной системы позиционирования (Global Positioning System, GPS) и инерциальной навигационной системы (Inertial Navigation System, INS) – обеспечивает функциональную независимость и устраняет многие проблемы отказов общего режима, которые иначе могли бы возникнуть:

Примеры: функциональная независимость

- **Замедление на земле.** Колесные тормоза, реверсоры тяги двигателей и наземные спойлеры.
- **Управление направлением на земле.** Управление носовым колесом, дифференциальное торможение и руль направления на высокой скорости.
- **Управление воздушным судном в воздухе.** Поверхности управления полетом и вектор тяги.
- **Определение относительного положения воздушного судна.** Связь и навигация.
- **Навигация.** Глобальная система позиционирования (Global Positioning System, GPS) и инерциальная опорная система (Inertial Reference System, IRS).
- **Определение угла атаки (Angle of Attack, AOA).** Флюгерный датчик и синтетический угол атаки, вычисленный по воздушной скорости и инерциальным данным.
- **Определение количества топлива.** Расход топлива двигателя и датчики уровня топлива в баках.

Независимость разработки элементов аналогично гарантирует, что элементы, связанные с функцией, не испытывают общих ошибок. Например, использование разных операционных систем в независимых элементах обеспечивает независимость разработки элементов и предотвращает ситуацию, которая могла бы возникнуть, если бы единая операционная система содержала неустраненную ошибку:

Примеры: независимость элементов

- Разные технологии: гидравлическая или электрическая энергия.
- Разные операционные системы реального времени.
- Разные языки программирования и ПО.
- Разные микропроцессоры.
- Разные коллективы и процессы.

Определение требований

Уровни гарантии разработки определены, и настало время определять требования. Что? Разве требования не учитываются в процессе назначения уровня гарантии разработки? Обычно требования уже известны или, по крайней мере, начинают формироваться параллельно с назначением уровня гарантии разработки, но уровень гарантии разработки связан с безопасностью и потому должен инициироваться без формального рассмотрения требований. Соображения безопасности должны предшествовать функциональности. Требования определяются отчасти для удовлетворения безопасности, установленной в

процессе назначения уровня гарантии разработки. Следовательно, определение уровня гарантии разработки предшествует определению требований. Как определяются требования? Через рассмотрение разных типов требований и того, как каждый из них относится к разрабатываемому воздушному судну или системе. Следующий рисунок показывает основные типы требований, которые должны быть определены, а также обычную, но не обязательную последовательность:

Основные типы требований к воздушному судну/системе

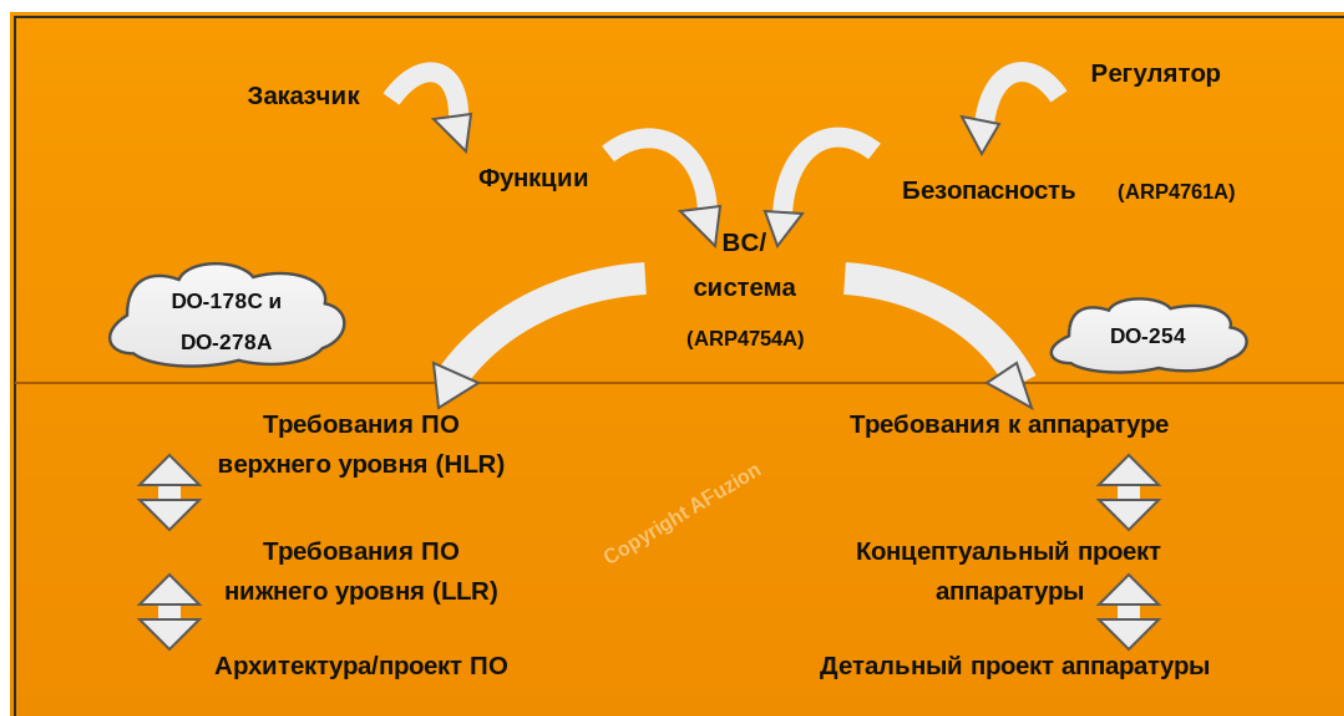
1. **Требования по безопасности.** Учитывают функциональные отказные состояния, целостность и готовность.
2. **Функциональные требования.** Учитывают основную функциональность: что делает воздушное судно или система.
3. **Эксплуатационные требования.** Описывают измеримые взаимодействия между летным экипажем и системой.
4. **Требования к интерфейсам.** Учитывают электрические входы/выходы и сигналы: источник, приемник, поведение.
5. **Требования к характеристикам.** Определяют измеримые характеристики: точность, разрешение, диапазон, время и так далее.
6. **Производные требования.** Документируют решения, не связанные с другими требованиями; часто относятся к безопасности.

Какие из перечисленных выше типов требований важны? Все. Но если серьезно, какие из них важнее всего? Рассмотрим. Функциональные характеристики и эксплуатационные характеристики важны, поскольку воздушное судно и системы должны выполнять свои цели, а эти цели должны быть четко определены; требования и дают такое определение. Эксплуатационные аспекты и интерфейсы, которые влияют на способность выполнять цели, также важны и должны быть определены. Однако безопасность имеет первостепенное значение, следовательно, и требования безопасности должны быть первостепенными. Но требования безопасности обычно труднее всего определять, поскольку здесь больше субъективности: насколько безопасно – это “безопасно”? Необходимо тщательно рассматривать артефакты оценки безопасности, чтобы учитывать соответствующие отказные состояния, режимы и эффекты. Знания должны быть сосредоточены на предотвращении, но мудрость должна преобладать, понимая, что стопроцентное предотвращение невозможно. Поэтому архитектурные аспекты, включая резервирование, разнородность, обособление и т. д., должны быть добавлены для поддержки требуемого уровня гарантии разработки и определены через требования. Подход к фиксации требований должен быть определен заранее, на этапе планирования. Почему? Планы оцениваются независимо до начала описанных в них мероприятий, поэтому запланированный процесс фиксации требований можно оценить до его запуска. Некоторые говорят, что самая большая ошибка на этом этапе состоит в том, что авторы требований ошибочно думают: “но пилот никогда бы так не сделал...”. Особенно по мере

того как кабины экипажа становятся все более автоматизированными, проявляется парадокс безвыходной автоматизации: чем больше автоматизации, тем больше пилот зависит от автоматизации, а значит, тем больше автоматизации...

Источники требований и трассируемость

В авиационной экосистеме разработки именно требования являются основой успеха. Когда Фред Брукс написал “Мифический человеко-месяц” в 1975 году, ни одного из этих руководств по воздушным судам, системам, безопасности, аппаратуре или ПО еще не существовало; они появились лишь несколькими годами позже. Но монументальная книга Брукса о сложных вычислительных системах привела к важным выводам, которые позже были встроены в формирующиеся авиационные руководства. Брукс отмечал, что главным источником ошибок в таких вычислительных системах являются предположения, в том числе предположения, которые инженеры делают при отсутствии определенных требований. Помните: верификация – это оценка соответствия реализации требованиям, тогда как валидация – это оценка самих требований. Валидация важнее, потому что если требования не являются корректными, полными и недвусмысленными, уже не имеет значения, что позже вы их фактически “верифицировали”. Все это подробнее рассматривается ниже и в последующих главах. Поэтому ARP4754A требует процесса управления требованиями, и затем этот процесс передается ниже по потокам процессов аппаратуры и ПО, как показано на следующем рисунке:



Источники требований и взаимосвязи в авиационной экосистеме

Что касается безопасности и требований, часто неправильно понимается роль производных требований. Что такое производные требования? Осторожнее с лингвистикой. Если английский не является вашим родным языком, не отчаивайтесь: носители английского, похоже, еще хуже понимают смысл “производных требований”,

потому что они выводятся из стремления инженера охватить иначе не задокументированную или упущенную возможность. Производные требования обычно не трассируются к родительскому требованию и не описывают очевидную функциональность.

Вместо этого производные требования представляют собой дополнительные аспекты, которые разрабатываемая система или воздушное судно должны реализовать, обычно для обеспечения безопасности, повторного использования или верифицируемости. Например, частоты выборки, использование обособления или добавление резервирования с мониторингом работоспособности и голосованием – все это примеры функциональности, обычно задаваемой через производные требования. Поскольку производные требования часто напрямую относятся к безопасности или закрывают пробел в определении функциональности, ARP4754A требует от разработчиков обеспечить обратную связь с процессом безопасности для всех производных требований. Любое изменение производных требований должно передаваться обратно в процесс безопасности, чтобы определить любое прямое или косвенное влияние на безопасность. Разработчики обязаны доказать обязательное применение этого процесса ко всем производным требованиям, что проверяется персоналом по гарантии качества при аудите.

Валидация требований

Поздравляем: требования к воздушному судну и системе зафиксированы, и пришло время начинать проектирование, а затем реализацию. Но вы умны и знаете, что позже будете верифицировать эти требования, чтобы убедиться, что реализация им соответствует. Подождите: откуда вы знаете, что у вас “правильные” требования? Именно. В этом и состоит роль валидации требований, и она должна предшествовать проектированию/реализации. Почему? Требования – это фундамент воздушных судов и систем. Как и при строительстве дома, изменение фундамента после начала строительства нежелательно по многим причинам, важнейшая из которых – безопасность. Действительно, необходимо выявить противоречивые, неполные или некорректные требования задолго до начала разработки. ARP4754A требует формального процесса валидации, в рамках которого требования оцениваются на корректность и полноту до того, как будут использованы в проектировании/реализации. Однако валидация требований выполняется на протяжении всего процесса разработки, потому что изменения неизбежны, и любые изменения требований или всего, что влияет на требование, должны проходить оценку через валидацию. Подход к валидации требований должен быть определен заранее в планах. Обычно используется отдельный план валидации, хотя нередко планирование валидации объединяют с планом верификации, поскольку валидация и верификация связаны.

Помните: требования к воздушному судну, системе и аппаратуре могут и должны проходить валидацию. Однако поскольку “валидация” включает полноту и корректность, требования к ПО (рассматриваемые далее в этой книге) НЕ валидируются: для оценки полноты ПО нужен как аппаратурный, так и системный контекст.

Валидация требований – это формальный процесс, а в авиации большинство формальных процессов требуют контрольных перечней. Даже пилот с налетом 5000 часов и огромным опытом формально обязан следовать контрольному перечню при многих ключевых действиях, включая взлет и посадку. И поскольку безопасность столь важна, на коммерческих воздушных судах используется второй пилот для дополнительной независимой верификации. К счастью, ARP4754A в разделе 5.4.3 дает критерии валидации требований высокого уровня, которые следует использовать как основу контрольного перечня валидации. Ради краткости они здесь не повторяются: в валидации требований в авиации мало что уникально по сравнению с другими критичными для безопасности отраслями, хотя сам термин “валидация” используется иначе – см. соответствующую главу этой книги о требованиях в авиации. Применяются обычные критерии оценки: корректность, недвусмысленность, отсутствие конфликтующей логики, идентифицируемость, трассируемость, влияние на безопасность и т. д.

Следующий рисунок содержит практические замечания по валидации требований к воздушному судну и системе:

Валидация требований к воздушному судну и системе: практические замечания

- Выявить интерфейсы, которые должны быть учтены в требованиях: с воздушным судном, другими системами, системными элементами и т. д.
- Зафиксировать эти интерфейсы в согласованных документах, например в техническом задании, плане, руководстве или документе требований.
- В таком документе должно быть описано поведение интерфейса при получении входных данных.
- Независимый рецензент должен вместе с автором требований критически проверить исходные допущения и трактовки зафиксированных требований.
- Элементы, не являющиеся сложными, могут считаться удовлетворяющими строгости уровня гарантии разработки элемента A (Item Development Assurance Level, IDAL A), если уверенность в них полностью обеспечивается сочетанием испытаний и анализа.

Степень строгости валидации, конечно, меняется в соответствии с критичностью, то есть с уровнем гарантии разработки. Таблица ниже, перепечатанная из документа SAE ARP4754A, ясно это объясняет. Обратите внимание, что “Р” в таблице, как указано, означает “рекомендуется”. Однако на практике большинство сертифицирующих органов по всему миру придерживаются более консервативного толкования и понимают это как “требуется”. По опыту автора органы сертификации почти всегда требуют применения рекомендуемых методов валидации.

Методы и данные (см. 5.4.6.a-f и 5.4.7)	Уровень гарантии разработки – А и В	Уровень гарантии разработки – С	Уровень гарантии разработки – D	Уровень гарантии разработки – E
Предварительная оценка безопасности воздушного судна/системы (PASA/PSSA)	P	P	Согл.	Н
План валидации	P	P	Согл.	Н
Матрица валидации	P	P	Согл.	Н
Сводка валидации	P	P	Согл.	Н
Трассируемость требований (непроизводные требования)	P	P	Согл.	Н
Обоснование требований (производные требования)	P	P	Согл.	Н
Анализ, моделирование или испытание	P	Одно из трех рекомендуется	Согл.	Н
Сходство (опыт эксплуатации)	Согл.	Одно из трех рекомендуется	Согл.	Н
Инженерное рассмотрение	P	Одно из трех рекомендуется	Согл.	Н

Примечание. В таблице: Р – рекомендуется для сертификации; Согл. – по согласованию для сертификации; Н – не требуется для сертификации.

Для каждого требования следует определить и затем применить сочетание рекомендуемых и допустимых методов, необходимое для установления требуемой уверенности в валидации этого требования.

Верификация реализации

Планы выполнены безупречно, требования кажутся идеально валидированными, а персонал по гарантии качества проводил аудит мероприятий на всем протяжении инженерного процесса, включая все переходы через реализацию. Готовы начинать летные испытания? Не так быстро. Помните: в авиации вероятность ошибки ПО принимается равной “1”, то есть ошибки случаются. Можно ли их обнаружить и снизить их последствия? В инженерной деятельности действует та же философия. Хотя хорошие планы, процессы и инженерная дисциплина дают измеримо лучшие системы с меньшим числом ошибок, верификация все равно требуется. Распространено мнение, что верификация повышает качество за счет обнаружения ошибок. Это не совсем неверно, но верификация на самом деле выполняется для оценки адекватности предшествующих мероприятий в сочетании с обратной связью, улучшающей эти мероприятия и саму реализацию. Поэтому авиационная верификация важнее и шире, чем простое обнаружение дефектов.

Обычно считается, что верификация выполняется исключительно для выявления ошибок с последующим их исправлением. Это верно лишь отчасти: верификация на уровне воздушного судна и авиационных систем по ARP4754A имеет более широкий охват, поскольку она также используется для оценки полноты и качества предшествующих мероприятий разработки, включая безопасность, требования и проектирование, а не только реализацию. Помните приведенное ранее предельно упрощенное уравнение верификации: “*верификация = рассмотрения + испытания + анализ*”.

Как обязательные, так и потенциальные мероприятия верификации более разнообразны. Степень строгости верификации возрастает с уровнем гарантии разработки: уровень А является наиболее интенсивным. Следующий рисунок показывает типичные виды верификации, применяемые в рамках ARP4754A:

Типичные виды верификации по рекомендуемой аэрокосмической практике ARP4754A (Aerospace Recommended Practice, ARP)

1. **Инспекция или рассмотрение.** Визуальная проверка артефактов на соответствие требованиям.
2. **Оценка с применением моделей.** Изучение моделей или их элементов для оценки поведения системы в реальном времени.
3. **Анализ.** Детальное исследование инженерных данных, как правило, по ожидаемым критериям приемлемости.
4. **Испытание или демонстрация.** Работа системы для сопоставления фактических результатов с ожидаемыми.
5. **Подобие или история эксплуатации.** Использование ранее полученного зачета верификации для идентичных или почти идентичных систем.

Как и валидация, описанная в предыдущем разделе, строгость верификации меняется в зависимости от уровня гарантии разработки. Это полностью описано в следующей таблице, перепечатанной из документа SAE ARP4754A:

Methods & Data	Development Assurance Level				
	A	B	C	D	E
Verification Matrix	R	R	R	A	N
Verification Plan	R	R	R	A	N
Verification Procedures	R	R	R	A	N
Verification Summary	R	R	R	A	N
ASA / SSA	R	R	R	A	N
Test	R	R	R (1+)	A	N
Inspection, Review or Analysis	R (1+)	R (1+)		A	N
Test – unintended function	R	R	A	A	N
Service Experience	A	A	A	A	A

R – Recommended for certification, A – As negotiated for certification
N – Not required for certification

Процессы поддержки воздушного судна/системы по ARP4754A

Управление конфигурацией (Configuration Management, CM) и гарантия процесса (Process Assurance, PA)

Ваши инженеры умнее Эйнштейна и организованнее отряда специалистов по эффективности; вы с первой попытки построили первую в мире идеальную авиационную систему. Браво. Сертификация наверняка пройдет легко, верно? Едва ли. Возможно, ваши инженеры действительно были блестящими, но что насчет более слабого звена, которое наймут позже для внесения изменений в систему? Именно поэтому необходимо профессиональное управление конфигурацией. И как вы докажете, что ваша стая Эйнштейнов действительно следовала установленному процессу? Независимая гарантия процесса должна проводить аудит работы инженеров и артефактов, чтобы оценивать соответствие установленным процессам. Управление конфигурацией и гарантия процесса – это поддерживающие процессы: их выходные данные не вносят непосредственного вклада в само воздушное судно или систему, но они совершенно необходимы.

Управление конфигурацией. Управление конфигурацией должно быть определено в плане, а затем выполняться согласно этому плану на протяжении всего срока жизни воздушного судна или системы, а не только до первоначальной сертификации. По сути, управление конфигурацией обеспечивает возможность знать и контролировать происхождение всех элементов, составляющих каждую поставленную конфигурацию воздушного судна или системы, а также соответствующих артефактов. Надлежащее управление конфигурацией налажено тогда, когда все могли бы покинуть проект, а последующий персонал все равно смог бы точно воспроизвести любую поставленную версию воздушного судна или системы и узнать полную историю всех участников, дат, версий и связей с другими версиями, будучи уверенным, что в эти версии не были внесены несанкционированные изменения. Следовательно, процессы управления конфигурацией должны обеспечивать установление базовых версий, управление

изменениями, архивирование и извлечение, а также поддержание безупречного указателя конфигурации, в котором перечислен каждый артефакт, использованный при проектировании системы или воздушного судна. Всегда помните: чем сложнее проект или чем изменчивее требования, тем выше риск ошибок управления конфигурацией, последствия которых почти невозможно устранить без чрезмерных и неоправданных усилий.

Гарантия процесса. Ранее гарантия процесса уже кратко затрагивалась, когда обсуждалось различие между аудитами и рассмотрениями. Помните: инженеры выполняют технические рассмотрения; гарантия процесса выполняет аудиты, ориентированные на процесс. И снова: в коммерческой и традиционной военной практике роль гарантии процесса отличается от ее роли в ARP4754A и документах серии DO. В экосистеме разработки авионики у гарантии процесса особые обязанности: она утверждает планы, стандарты и контрольные перечни, а затем оценивает соблюдение этих документов. Она не выполняет рассмотрения – она выполняет аудиты, включая оценку критериев перехода. Планы разработки авионики задают основные мероприятия через входные и выходные артефакты для каждого такого мероприятия. Инженерный персонал должен работать с этими входными и выходными артефактами, а соблюдение этого порядка оценивается в ходе инженерных рассмотрений. Рассмотрения выполняет инженерный персонал, используя инженерные контрольные перечни. Гарантия процесса оценивает совокупность этих входных артефактов, выходных артефактов и проверок по инженерным контрольным перечням как переход и тем самым определяет, были ли выполнены критерии перехода. Аудиты выполняет гарантия процесса, используя свои контрольные перечни.

Процессы управления конфигурацией и гарантии процесса должны быть определены в плане. Для управления конфигурацией независимость не обязательна: инженеры могут выполнять его собственными силами, хотя в крупных организациях обычно есть команды или сотрудники по управлению конфигурацией, отвечающие только за это. Однако гарантия процесса должна быть доказуемо независимой от инженерного подразделения. Участники гарантии процесса не должны подчиняться руководителю по системной инженерии или инженерии воздушного судна; они должны подчиняться через отдельную и независимую организацию гарантии качества, чтобы обеспечить демонстрируемую независимость.

Теперь общие системные процессы ARP4754A описаны. В последующих главах этой книги обсуждаются дополнительные руководящие указания для бортового ПО и аппаратуры, баз данных, наземных систем и практически любого другого подмножества воздушных судов и систем, связанного с инженерным качеством или безопасностью. Для каждого из этих объектов существуют специальные руководящие документы по соответствующей области, но в целом они повторяют базовые процессы, заданные в ARP4754A: безопасность, планирование, требования, реализация, верификация и валидация, управление конфигурацией и гарантия качества. Постройте хороший фундамент воздушного судна и системы, и остальная часть здания тоже может быть

прочной. Но помните: любой может переплатить за индивидуальные решения и при избытке времени или графика добиться приемлемого результата. Мудрые достигают лучших результатов быстрее и экономичнее, если заранее знают свои цели и планируют соответственно.

DO-178C: вводная глава о ПО авионики

Автор: Вэнс Хилдерман

Рецензенты:

- Гамзе Эроглу, бывший старший специалист по сертификации, STM
- Джордж Мейер, представитель Федерального управления гражданской авиации США (Federal Aviation Administration, FAA) по инженерной экспертизе (Designated Engineering Representative, DER)
- Билл Поттер, менеджер по техническому маркетингу, MathWorks

К 2005 году DO-178B, руководящий документ по ПО авионики под названием *“Вопросы ПО при сертификации бортовых систем и оборудования”*, уже достиг солидного возраста, и его просто пора было обновить. Что могло быть проще? Правда, мир разработки ПО, даже в обычно консервативном сообществе авионики, за это время ушел далеко вперед. DO-178B, опубликованный пятнадцатью годами ранее, никак не мог предвидеть многие из этих изменений. В технологиях, как и в жизни, постоянны только изменения. Особенно в программных технологиях, где изменения измеряются месяцами, а не десятилетиями.

Как отмечалось в предыдущих главах, DO-178 для ПО авионики, или его идентичный европейский аналог ED-12, где ED означает «европейский документ» (European Document), является частью общей экосистемы авиационной разработки и сертификации. Теоретически и в нормальном порядке разработка ПО авионики следует за другими ключевыми процессами: 1) обеспечением безопасности воздушного судна и систем, 2) разработкой воздушного судна и систем и 3) разработкой аппаратуры. Но реальность часто отличается от теории, и здесь именно такой случай: первая версия DO-178 была опубликована в 1980 году, задолго до появления других руководств по безопасности, разработке воздушного судна и систем, а также аппаратуре. Поэтому DO-178 оказался не простым винтиком авиационного механизма, а исторической и методической основой для последующих руководств и стандартов. Неудивительно, что множество авиационных руководств, выпущенных после DO-178, выглядят знакомо: они намеренно воспроизводят его подход, что в итоге поддерживает единообразие.

В человеческой и мировой истории конкретные даты редко важны сами по себе; важнее окружающий контекст. С DO-178 то же самое. Напомним: системы, доставившие людей на Луну в середине 1960-х, были разработаны без какого-либо DO-178. Однако у тех ракет и лунных систем были сравнительно небольшие кодовые базы, созданные исключительно сильными инженерами без особой оглядки на расширяемость. Кроме того, программе

“Аполлон” не приходилось в такой же мере учитывать средние или слабые навыки пилотов, неподготовленных пассажиров, экстремальную погоду, предотвращение столкновений и тому подобное. Но к середине 1970-х быстрый рост компьютеризации и распространенности разработки в гражданской авиации потребовал проверяемой структуры согласованных процессов разработки ПО авионики. Номер DO-177 уже был занят 177-м документом серии DO-XXX, поэтому следующему доступному номеру, 178, досталось это зарождающееся руководство. Ниже приведена слишком краткая история DO-178:

Сильно упрощенная история развития DO-178 и его контекст

Документ	Год	Основа	Темы
DO-178	1980-1982	Зарождающиеся основы ПО	Артефакты, документы, трассируемость, тестирование
DO-178A	1985	DO-178	Процессы, тестирование, компоненты, четыре уровня критичности, рассмотрения, каскадная методология
DO-178B	1992	DO-178A	Интеграция, критерии перехода, различные методы разработки, данные (не документы), инструменты
DO-178C	2012	DO-178B	Снижение субъективности; учет моделирования, детальных требований, объектно-ориентированной технологии и формальных методов: “экосистема”

Важно помнить исторический контекст: к концу 1970-х, когда потребность в руководстве по ПО авионики стала очевидной, настоящая инженерия ПО еще только формировалась. Информатика и вычислительная техника развивались, но процесс разработки крупного или сложного ПО все еще оставался неформальным. Книга Брукса “Мифический человеко-месяц” обобщила трудности такой разработки и указала на практические меры: более качественные требования к ПО и более четкую коммуникацию, чтобы уменьшать число допущений. Так появился DO-178. Как и конституции успешных стран, руководства для быстро развивающейся области, такой как ПО и авиация, лучше работают тогда, когда делают акцент на гибкости, а не на подробных рецептах. Поэтому DO-178 не объяснял и до сих пор не объясняет, как именно достигать целей. Вместо этого DO-178 описывает цели и показывает, что с ростом критичности ПО, то есть уровня гарантии разработки (Development Assurance Level, DAL), должно выполняться все больше таких целей. В сегодняшнем DO-178C максимум составляет 71 цель для ПО уровня А. Авторами DO-178 были в основном инженеры, работавшие у крупнейших мировых разработчиков и изготовителей авионики. Boeing, Airbus и многие другие изготовители планеров проектируют и интегрируют воздушные суда, но не являются основными разработчиками ПО. В отличие от многих других стандартов, связанных с безопасностью и обычно создаваемых независимыми органами стандартизации или государственными специалистами, DO-178 уникален тем, что его разрабатывали коммерческие инженеры из тех самых компаний, которым затем предстояло по нему работать.

Перематываем на 25 лет вперед ...

Из-за непрерывного развития инструментов и методов разработки ПО стратегии сертификации авионики тоже нуждались в обновлении. Эти вопросы можно было закрывать консультативными циркулярами Федерального управления гражданской авиации США (Advisory Circular, AC) или даже позиционными документами Группы сертификационных органов по ПО (Certification Authorities Software Team, CAST). Собственно, консультативные циркуляры и документы CAST постоянно выпускались именно для этого. Конечно, было бы намного быстрее и проще выпускать дополнительные документы, чем обновлять сам DO-178B. По сути, сообществу разработчиков авионики нужно было выбрать одну из четырех стратегий будущей разработки и сертификации авионики:

1. Ничего не делать.
2. Слегка подправить DO-178B.
3. Существенно доработать DO-178B.
4. Полностью переработать DO-178B.

Вариант “ничего не делать” явно не годился: развитие технологий и опытные разработчики, часть которых все чаще умела обыгрывать систему и использовать лазейки, означали, что DO-178B устарел. Полная переработка DO-178B тоже была не нужна. В отличие от крупного пересмотра при переходе от DO-178A к DO-178B, подавляющее большинство текста DO-178B, более 98 %, считалось пригодным, поэтому капитальный ремонт не требовался. Оставались варианты 2 и 3. В зависимости от точки зрения обновление DO-178B до DO-178C, изменившее буквально 2 % текста DO-178B, можно считать и небольшим, и крупным. Если смотреть только на довольно скромные изменения в самом DO-178B, обновление до DO-178C было совсем небольшим. Если же смотреть на экосистему четырех новых дополнений, добавленных к DO-178C и связанным авиационным руководящим материалам, совокупные изменения были весьма крупными. Эти дополнения кратко описаны далее в этой главе и отдельно рассматриваются в других главах книги.

Чтобы вернуть DO-178B актуальность, нужно было решить две задачи:

1. Дать руководство по приемлемому применению новых технологий инженерии ПО, включая современные программные инструменты, разработку на основе моделей (Model-Based Development, MBD), объектно-ориентированную технологию (Object-Oriented Technology, OOT) и формальные методы (Formal Methods, FM).
2. Уменьшить возможность для опытных разработчиков пользоваться лазейками, одновременно заявляя о соответствии DO-178B.

Первую задачу, то есть руководство по приемлемому применению новых технологий, можно было легко решить через консультативные циркуляры или документы CAST, не обновляя DO-178B. Но вторая задача, связанная с опытными или просто недостаточно информированными разработчиками, которые использовали определенные лазейки в DO-178B, была сложнее и потребовала конкретных изменений в самом DO-178B. Добро пожаловать в следующую версию, которая теперь перед нами: DO-178C.

Для критически важного с точки зрения безопасности ПО надежность является очевидным требованием. В авиации надежность фактически рассматривается через призму безопасности: формальный анализ безопасности выполняется до разработки собственно ПО и продолжается во время разработки. В авионике стратегия безопасности строится на рекомендуемой аэрокосмической практике SAE ARP4761A (Aerospace Recommended Practice, ARP). См. предыдущую главу о SAE ARP4761A. Однако в авиации само ПО не анализируют с точки зрения безопасности: оценки безопасности выполняются на уровне воздушного судна, системы и аппаратуры, но не на уровне ПО. Почему? Просто: “вероятность отказа ПО равна 1, то есть 100 %”. В других отраслях действительно прилагают максимум усилий, чтобы выполнять оценки безопасности проекта ПО и кода. Но не в авиации. Проще говоря, оценки безопасности, выполняемые непосредственно для ПО, качественны и субъективны, а не количественны и детерминированны. При этом проектирование и реализация ПО напрямую связаны с безопасностью. Вместо того чтобы предписывать какую-либо оценку безопасности самого ПО, авиация задает процессы инженерии ПО, а затем требует верификации и аудита этих процессов, чтобы доказуемо подтвердить их соблюдение. Но оценки безопасности ПО как таковой не выполняются. Лучшее, что можно сделать на уровне ПО, – выполнить все 71 цель DO-178C; это должно примерно соответствовать эквивалентному фактическому уровню безопасности, который следует из оценок безопасности системы и аппаратуры. Эти 71 цель в DO-178C называются целями, а не субъективными оценками, потому что их можно оценивать однозначно и детерминированно.

Вокруг возможности явно определить фактическую надежность, или “безопасность”, ПО существует немало мифов; как обычно с мифами, вымысла в них больше, чем реальности. Но хорошие истории мы любим. Такая история о безопасности ПО могла бы утверждать, что ПО можно аналитически исследовать и определить, насколько оно безопасно или надежно. Отчасти это верно: там, где есть алгоритмы или модели, мы в разной степени можем математически установить или предсказать поведение ПО. Однако предельная цель, то есть настоящая мера способности предсказать надежность ПО, сводилась бы к утвердительному ответу на вопрос: “Могу ли я доказать, что это ПО совершенно?” Почти всегда, кроме самых простых программ, ответ будет: “нет”. Почему? Потому что ПО настолько хорошо, насколько хорошо его самое слабое звено. Задумайтесь: как назвать ПО, которое совершенно на 99,9 %? Очевидно, “несовершенным на 0,1 %”. Тогда какова в авионике предполагаемая вероятность, что это 0,1 % несовершенства проявится как аномалия в полете? Почти стопроцентная.

Следовательно, мы не можем доказуемо утверждать, что ПО совершенно. Поэтому авиация применяет другой подход. Поскольку мы не можем легко доказать, что наше ПО авионики совершенно, вместо этого мы делаем следующее:

Подход к ошибкам в ПО авионики

1. Определить строгий процесс разработки по DO-178 и следовать ему: чем выше критичность, тем выше строгость.
2. Максимально предотвращать ошибки.
3. Признать, что невозможно предотвратить все ошибки, поэтому ввести обнаружение ошибок.
4. Принять: обнаружение ошибок хорошо; снижение последствий ошибок лучше.

С момента выпуска DO-178B в начале 1990-х наши знания о процессах, методах и стратегиях разработки критически важного с точки зрения безопасности ПО значительно выросли. Если бы сегодня перед нами была та же предметная область, что и двадцать лет назад, мы почти могли бы гарантировать сокращение числа ошибок ПО более чем на 99 %. Не верите? Я достаточно стар, чтобы помнить автомобильные поездки с родителями в 1960-е и 1970-е годы.

Повсюду стояли неисправные автомобили: спущенные шины, перегретые радиаторы, сломанные стартеры. А сегодня? Большинство людей не знает, как заменить спущенное колесо, справиться с перегретым радиатором или “прикурить” автомобиль. Производители лучше поняли автомобильную предметную область и значительно повысили надежность этих массовых компонентов. Обычно проблемы в автомобилях создают именно более новые и менее зрелые технологии.

Теперь сравните это с авионикой: размер и сложность ПО авионики выросли экспоненциально. Очень немногие авионические системы, содержащие ПО, остались неизменными. Все чаще ПО служит и отличительным признаком продукта, и средством преодоления технических барьеров. От ПО требуют делать больше при более коротких сроках разработки. Решение? Принять новые технологии разработки ПО и новые подходы к его исполнению. Однако DO-178B не мог предвидеть эти новые подходы к разработке и исполнению ПО. Поэтому коммерческий мир принял новые технологии быстро, а авионика двигалась следом медленно.

Мировые сертифицирующие органы, такие как Агентство Европейского союза по авиационной безопасности (European Union Aviation Safety Agency, EASA) и Федеральное управление гражданской авиации США, укомплектованы умными и трудолюбивыми людьми, но обычно у них меньше практического опыта недавней разработки ПО с применением этих новых технологий. Их одобрение коммерческой авионики – палка о двух концах. Если они не одобряют систему, авионика не полетит, а инновации и решение сложных задач авионики будут сдержаны. Но одобрение означает, что эти системы признаются безопасными теми же людьми, на которых частично ляжет ответственность за

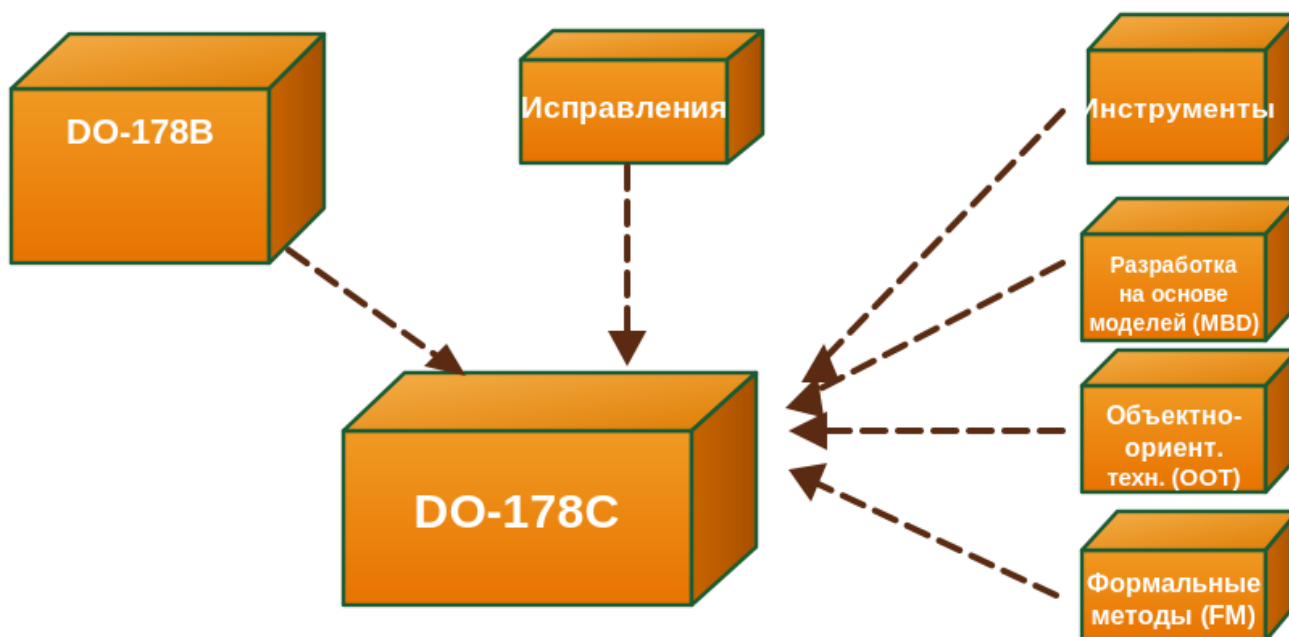
последующие отказы. Результат – настоящая дилемма и непростое положение для органа сертификации авионики. Но эти органы видели приближающуюся необходимость принять новые технологии и не спали на посту. Они создавали группы для изучения новых разработок и выработки стратегий их внедрения. Они привлекали отраслевых экспертов и технические площадки, чтобы собирать и распространять информацию. Они писали документы по проблемным вопросам. Они координировали действия через океаны. Они обучали свой персонал и отрасль. Они адаптировались. И они понимали, что DO-178B не переживет продолжающуюся технологическую эволюцию в прежнем виде: нужна обновленная версия, DO-178C.

Какие именно технологические изменения потребовали обновить DO-178B? Компиляторы? Линковщики? Микропроцессоры? Инструменты тестирования ПО? Не совсем.

Эти продукты продолжали развиваться и улучшаться, но не они стали главным катализатором изменений в разработке или сертификации авионики. Потребность в DO-178C породили технологии, которые позволяли лучше управлять главной задачей разработки ПО авионики: растущей сложностью этого ПО.

Пятьдесят лет назад в самолетах не было никакого “ПО”. У “Аполлона-11”, запущенного в 1969 году как первого аппарата серии лунных посадок, был навигационный компьютер с 36 КБ постоянной памяти (постоянное запоминающее устройство, ПЗУ; Read-Only Memory, ROM) и 2 КБ оперативной памяти (оперативное запоминающее устройство, ОЗУ; Random Access Memory, RAM). Ваши наручные часы на батарейке или бытовая микроволновая печь почти наверняка мощнее. Через двадцать лет коммерческие воздушные суда уже имели более миллиона строк исходного кода. Сегодня воздушные суда имеют более 20 миллионов строк кода при намного большей интеграции и сложности. Линейный рост? Едва ли: экспоненциальный. Одновременно сближение военной и гражданской авионики вместе с возросшей потребностью в повторном использовании ПО означали, что нужны новые методы разработки ПО. В частности, разработчики авионики больше всего хотели использовать три технологии, чтобы лучше управлять меняющимся ландшафтом: 1) разработку на основе моделей; 2) объектно-ориентированную технологию; 3) формальные методы. Эти новые технологии, вместе с необходимостью устранить несколько слабых мест в DO-178B и добавить новые аспекты квалификации инструментов, привели к появлению DO-178C, как показано ниже:

Ключевые составляющие DO-178C:



Разработка на основе моделей, объектно-ориентированная технология и формальные методы применительно к разработке ПО авионики совсем кратко описаны ниже.

Краткое изложение разработки на основе моделей (MBD), объектно-ориентированной технологии (ООТ) и формальных методов (FM)

Краткое изложение разработки на основе моделей, объектно-ориентированной технологии и формальных методов

- **Разработка на основе моделей (Model-Based Development, MBD).** Использует формальные спецификации и модели проекта для описания системы; позволяет системным инженерам и инженерам ПО использовать один и тот же язык; обеспечивает автоматическую генерацию кода из модели; обычно реализуется с использованием унифицированного языка моделирования UML (Unified Modeling Language) или языка системного моделирования SysML (Systems Modeling Language).
- **Объектно-ориентированная технология (Object-Oriented Technology, OOT).** Разработка ПО с использованием объектов вместо последовательностей инструкций; повышает переносимость и повторное использование ПО за счет развитых программных конструкций и архитектурных решений; обычно реализуется на C++ или Ada.
- **Формальные методы (Formal Methods, FM).** Используют математические обозначения и анализ для определения всех аспектов заданной подсистемы; доказательство корректности формальных обозначений заменяет другие, обычные методы верификации; особенно полезны для сложной математической логики, менее пригодной для традиционной верификации по DO-178B.

Итак, разработка авионики явно осваивала современные инструменты, разработку на основе моделей, объектно-ориентированную технологию и формальные методы, поэтому требовалось сформулировать новое руководство для их безопасного внедрения. Фактически эти новые технологии не были напрямую добавлены в основной текст DO-178. Вместо этого были написаны совершенно новые дополнения: по одному для инструментов, разработки на основе моделей, объектно-ориентированной технологии и формальных методов. Такой подход выбрали по двум причинам:

1. Если оформить их как дополнения вне основного текста DO-178, другие авиационные руководства могли ссылаться на них и легко применять их. Например, документы серии DO-XXX: DO-254 (бортовая аппаратура), DO-278C (системы связи, навигации, наблюдения и организации воздушного движения; Communication, Navigation, Surveillance / Air Traffic Management, CNS/ATM), DO-297 (интегрированная модульная авионика) и другие.
2. Если бы все четыре дополнения были включены прямо в DO-178C, он вырос бы примерно в пять раз, до примерно 600 страниц, и читать его стало бы существенно труднее.

Помимо четырех технологических дополнений, изменения и добавления внесли и в сам основной текст DO-178; так он стал DO-178C. В отличие от пересмотра DO-178A в DO-178B, который двадцатью годами ранее был серьезным концептуальным сдвигом, изменения в основном тексте DO-178C относительно невелики. Они кратко представлены ниже:

Ключевые изменения в основном тексте DO-178C

Ключевые изменения в основном тексте DO-178C

1. Исправить незначительные недостатки формулировок.
2. Добавить новые цели для уровней А, В и С.
3. Ужесточить верификацию исходного кода по отношению к бинарному коду для уровня А.
4. Добавить новые сертификационные критерии для компонентов параметрических данных.
5. Повысить детализацию требований низкого уровня, чтобы обеспечить структурное покрытие напрямую через тестирование на основе требований.

В DO-178B действительно было несколько небольших несогласованностей формулировок, особенно если рассматривать его вместе с более поздними руководящими материалами, включая DO-248, DO-254, DO-278 и DO-297; в DO-178C большую часть этих несогласованностей исправили. Одна область, которую многие пользователи DO-178B надеялись увидеть более полно разъясненной, связана с термином “робастность”. К сожалению, этот термин в основном оставили без изменений, поэтому в

тестировании на робастность сохраняется субъективность, с которой каждому пользователю приходится разбираться самостоятельно. Действительно, областей и реализаций авионики так много, что невозможно заранее описать все возможные проявления необходимой робастности.

За несколькими заметными исключениями DO-178C не повысил существенно строгость, требуемую при разработке ПО авионики. В конце концов, в комитетах DO-178C предусмотрительно было широко представлено пользовательское сообщество, а оно не хотело чрезмерно увеличивать затраты, сроки или риски. Четкое разграничение между “критичным” ПО, то есть уровнями А, В и С, и “менее критичным” ПО уровня D сохранилось. Более того, разграничение между уровнями С и D даже усилилось: к уровню С добавили дополнительные цели, а сам уровень С приблизился к уровням А/В из-за необходимости подтверждать более полное покрытие структуры ПО при покрытии на основе требований.

Количественное изменение числа целей, требуемых для уровней DO-178C, показано ниже:

Уровень / отказное состояние	DO-178C Количество целей	DO-178B Количество целей	DO-178C Цели, требующие независимости
А / катастрофическое	71	66	33
В / опасное	69	65	21
С / значительное	62	57	8
Д / незначительное	26	28	5
Е / без последствий	0	0	0 (докажите, что у вас уровень Е!)

Примечание: для уровня Е достаточно доказать посредством оценки безопасности, что это действительно уровень Е.

DO-178C сохранил тот же базовый процесс планирования, разработки и обеспечения корректности, но эти процессы нужно обновить с учетом описанных выше изменений.

Методы и данные DO-178C по уровням гарантии разработки:

Методы и данные	Уровень А	Уровень В	Уровень С	Уровень D	Уровень Е
Матрица верификации	Р	Р	Р	П	Н
План верификации	Р	Р	Р	П	Н
Процедуры верификации	Р	Р	Р	П	Н
Сводка верификации	Р	Р	Р	П	Н
Оценка безопасности воздушного судна (Aircraft Safety Assessment, ASA) / оценка безопасности системы (System Safety Assessment, SSA)	Р	Р	Р	П	Н
Тестирование	Р	Р	Р	П	Н
Инспекция, рассмотрение или анализ	Р (1+)	Р (1+)	Р (1+)	П	Н
Тестирование: непредусмотренная функция	Р	Р	П	П	Н
Опыт эксплуатации	П	П	П	П	П

Р – рекомендуется для сертификации; П – по согласованию для сертификации; Н – не требуется для сертификации.

На этом вводная глава о DO-178C завершается. Дополнительные подробности приведены в следующих материалах. К этому моменту у читателя должны возникнуть несколько вопросов, в том числе:

1. *DO-178C сделал мою работу проще или сложнее?*
2. *Если я не использую разработку на основе моделей, объектно-ориентированную технологию или формальные методы, обязан ли я все равно придерживаться DO-178C и, если да, что именно для меня меняется?*
3. *Я модифицирую приложение по DO-178B, и его нужно сертифицировать на новой платформе по DO-178C. Как это повлияет на меня?*
4. *Если DO-178C не является первым в мире совершенным руководством по ПО, какие наиболее распространенные проблемные области и трудности мне следует заранее учитывать?*

Эти вопросы крайне актуальны, и ответ на каждый из них – “зависит от обстоятельств”. Человеческая потребность в черно-белых ответах остается неудовлетворенной, потому что сами вопросы слишком многоцветны. Но ответы есть, и в целом они обоснованны.

Авиационное ПО по DO-178C

Автор: Вэнс Хилдерман

Рецензент:

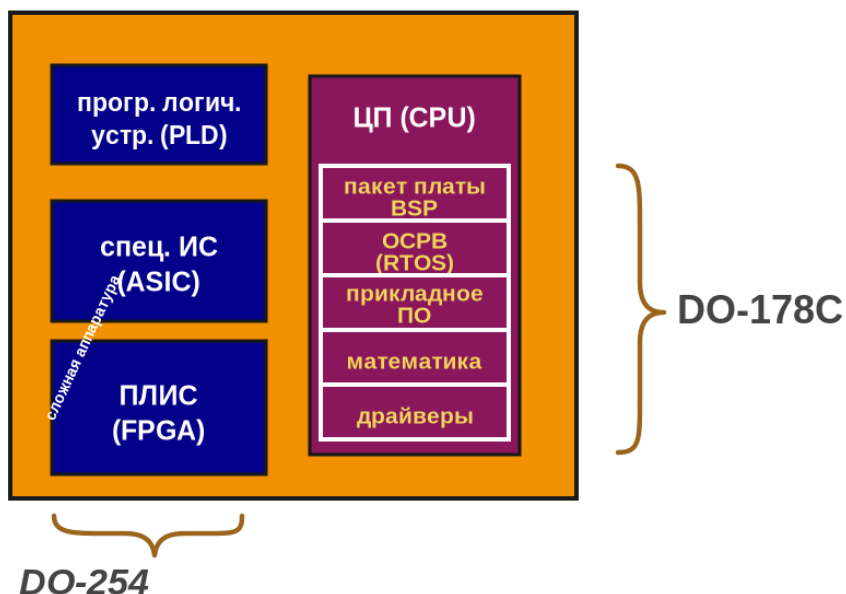
Г-н Франсуа Гуэй, представитель Федерального управления гражданской авиации США (Federal Aviation Administration, FAA) по инженерной экспертизе (Designated Engineering Representative, DER)

У DO-178 довольно безобидное название: *“Рекомендации по ПО при сертификации бортовых систем и оборудования” (Software Considerations in Airborne Systems and Equipment Certification)*. Но эту небольшую 144-страничную книгу лучше не оценивать по обложке. На практике “178”, как его называют в отрасли, во многом считается библией разработки авиационного ПО. Интересно, что с момента первой редакции в 1980-х годах, когда в системах, критичных для безопасности, ПО было сравнительно немного, этот документ стал фактическим воплощением подхода для многих таких систем. При внимательном изучении стандартов для военной авиации, медицинских изделий, железных дорог, автомобильной отрасли и атомной энергетики обнаруживается заметное сходство с 178. Случайность? Едва ли. Был ли 178 первым стандартом такого рода? Это почти неважно, за исключением одного: история имеет значение. Почему история важна для DO-178? Потому что 178 прошел три последующие редакции, и важно понимать, почему эти изменения понадобились.

Помните притчу о семи слепцах и слоне? Каждый касался своей части слона и пытался описать, к чему прикасается, ничего не зная об остальных частях. Задача заведомо невозможная, но в авиации в такую ловушку попасть легко: воздушные суда и их системы настолько сложны, что ни один человек не способен полностью понять все тонкости. Иными словами, без понимания общей картины мы немногим лучше тех слепцов. DO-178 пытается задать такую структуру, в которой разработчики и сертифицирующие органы могут работать с полным обзором и оставить остаточную слепоту позади. 178 действительно дает такую структуру. Но он не гарантирует ни идеальной ясности, ни тем более совершенства. Что, авиационное ПО не идеально? Конечно, нет. DO-178 не идеален? Тоже нет. Перефразируя знаменитую фразу Уинстона Черчилля о демократии: “DO-178 – худший стандарт в мире... если не считать всех остальных!”

Авиационные системы обычно строятся вокруг линейно-заменяемого блока (Line Replaceable Unit, LRU), где логика распределяется между ПО по DO-178C и аппаратурой по DO-254, как показано ниже:

Типовой линейно-заменяемый блок авионики:



В книге “Сертификация авионики: полное руководство по DO-178 и DO-254” (*Avionics Certification – A Complete Guide To DO-178 & DO254*) основной автор, то есть тот же автор, чью книгу вы сейчас читаете, потратил более 200 страниц на объяснение того, что такое DO-178. Коротко перечислить удобнее не свойства DO-178, а распространенные неверные представления о нем. Неверно считать DO-178:

- предписывающим и жестким стандартом для ПО;
- косным государственным стандартом, оторванным от рыночных реалий;
- техническим стандартом, устаревающим сразу после публикации;
- трактатом о разработке ПО;
- государственным стандартом, написанным людьми без серьезного практического опыта.

На самом деле DO-178 является противоположностью этого списка. Успех DO-178 обычно связывают с тем, что он:

- является гибкой структурой для разработки бортового ПО;
- способен охватывать почти все типы систем независимо от области применения и сложности, а также развиваться вместе с технологическим прогрессом;
- в значительной степени написан компетентными и успешными техническими специалистами, которые вместе со своими коллегами сами должны будут ему следовать, а значит заинтересованы в его полезности;
- не навязывает отрасли жесткий способ того, “как” разрабатывать и верифицировать ПО, а прежде всего сосредоточен на том, “что” требуется;
- задает минимально необходимый объем проверок и противовесов в зависимости от критичности бортового ПО, тем самым снижая стоимость соразмерно требованиям безопасности.

Чтобы начать понимать DO-178, нужно понимать “экосистемы”. Например, невозможно понять математический анализ, не понимая сложения, вычитания, уравнений и еще нескольких математических понятий: математический анализ – лишь часть математической экосистемы. Аналогично DO-178 является частью экосистемы разработки и сертификации авионики, куда входят процесс оценки безопасности по рекомендуемой аэрокосмической практике ARP-4761 (Aerospace Recommended Practice, ARP), разработка авионических систем по ARP4754A, гарантия разработки аппаратуры по DO-254, испытания на внешние воздействия и электромагнитные помехи (Electromagnetic Interference, EMI) по DO-160, а также многие другие важные аспекты. Во многом DO-178 похож на ногу того самого слона: это необходимая часть, позволяющая слону устойчиво стоять. Нога не гарантирует здоровье, но явно снижает вероятность преждевременной гибели...

DO-178 исходит из предпосылки, а для коммерческой авионики – из требования сертифицирующего органа, что пользователь применяет DO-178 лишь как одно звено в цепочке сертификации авионики. Авионика не будет ни безопасной, ни соответствующей требованиям, если до DO-178 нет безопасной основы. Что предшествует применению DO-178? Обычно разрабатывается план сертификации конкретного проекта (Project Specific Certification Plan, PSCP), который определяет экосистему авионики для данной авионической системы, включая применимость DO-178. Экосистема, указанная в PSCP, обычно включает формальную оценку функциональных опасностей (Functional Hazard Assessment, FHA) по ARP-4761, а затем определение системных требований к авионике по ARP-4754A. Что произойдет, если начать работы по DO-178, не учитывая эти предшествующие действия? Как правило, вы получите неплохую тренировку, после которой вам “посчастливится” вернуться и сделать все заново. DO-178 действительно требует безопасной основы с формально определенными системными требованиями, а утверждать обратное – безрассудство. Коротко говоря, DO-178 не проверяет, хороши системные требования или плохи. DO-178 предполагает, что системные требования корректны, и обеспечивает их адекватный перенос в ПО и верификацию соразмерно уровню разработки. Но качество авиационного ПО не может быть выше качества исходных системных требований.

DO-178 – сравнительно небольшой документ, намного меньше тех документов, которые вы создадите, чтобы доказать, что ему следовали. Говорят, что документы, используемые при разработке самолета, никогда не поместились бы внутри этого самолета. Точно так же документы, подготовленные для соответствия DO-178, гарантированно превысят по объему распечатанный исходный код. Почему? Потому что, как и в любой нормативной структуре для ПО, критичного для безопасности, нужны планы, стандарты, спецификации, проекты, рассмотрения, анализы, тестирование и аудиты, чтобы определить, оценить и доказать соответствие. То же самое верно для военной авиации, медицинских изделий, атомной энергетики, железных дорог и автомобильного ПО, критичного для безопасности.

После того как создана формальная структура для рассмотренных и управляемых анализов функциональных опасностей и системных требований, пора рассмотреть применение DO-178. С чего начать? Можно взять экземпляр DO-178 и сопутствующие документы и попытаться их понять. После нескольких прочтений вы, по крайней мере, убедите себя, что они интересны и важны. Но как действительно начать применять DO-178 и с чего начинать, когда кажется, что все строго, но ничего прямо не требуется? Вот здесь уже нужно по-настоящему понимать DO-178.

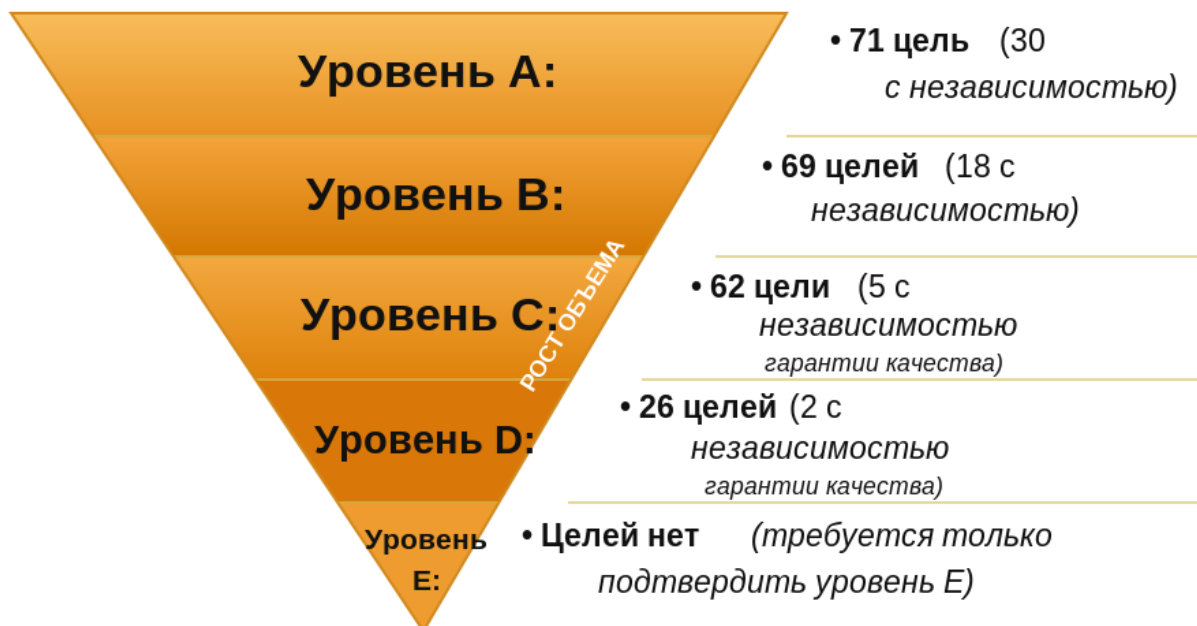
Уровни гарантии разработки (критичности).

Прежде всего нужно понимать уровень гарантии разработки (Development Assurance Level, DAL) ПО, которое вы разрабатываете по DO-178. Строгость планирования, разработки и подтверждения корректности ПО напрямую связана с его DAL, который часто называют “уровнем критичности”. Существует пять уровней, от уровня Е до наиболее строгого уровня А, как показано ниже:

Пирамида уровней критичности:

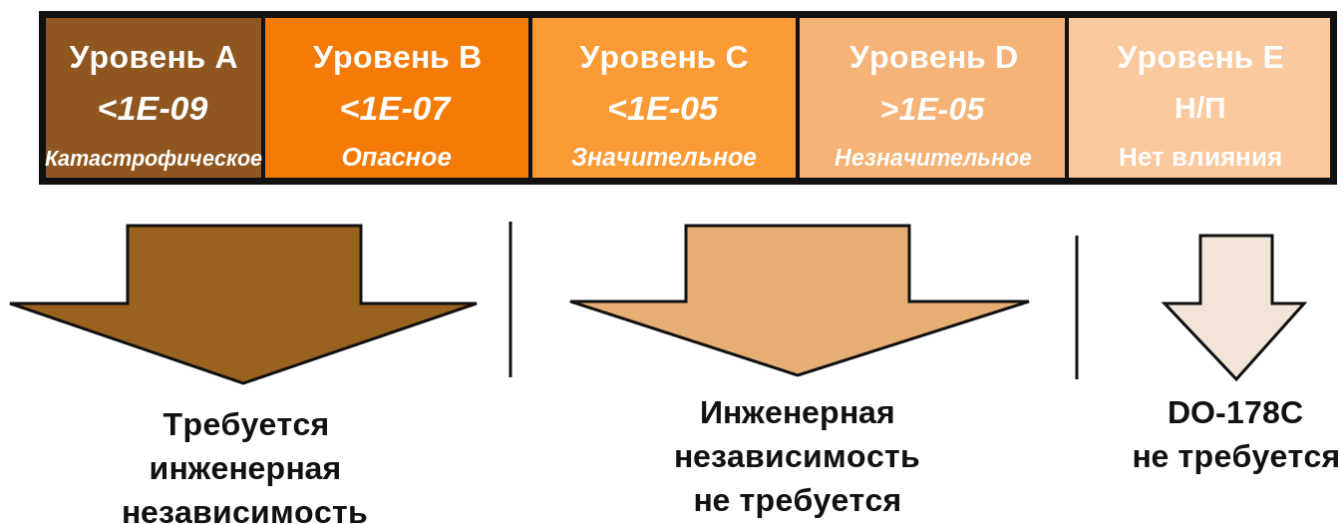


Помните: до начала разработки ПО DAL должен быть определен в рамках формального процесса оценки безопасности. Исходный DAL является выходом оценки функциональных опасностей воздушного судна и системной оценки функциональных опасностей. Позднее DAL может быть изменен в рамках непрерывного процесса оценки безопасности, который включает обратную связь и повторное рассмотрение со стороны инженеров по безопасности. Как отмечалось ранее, уровень инженерной строгости при разработке и верификации основан на DAL, как показано ниже:



Обратите внимание, что назначенный уровень и соответствующая надежность зависят от типа воздушного судна. Следующий пример относится к самолетам части 25 (14 CFR Part 25, правила летной годности FAA для самолетов транспортной категории), то есть к более крупным самолетам:

Инженерная “независимость” по уровню гарантии разработки (DAL). Гарантия качества всегда должна быть независимой



На рисунке выше показано, что “независимость” требуется для DAL В и в еще большей степени для DAL А. Речь идет о независимости верификации: верификацию выполняет другой человек по другому процессу. Такая независимость не означает другую компанию или другое место работы. На практике даже полезно, когда независимый верификатор находится рядом с разработчиками: так лучше используется экспертиза и сохраняются знания.

Можно ли просто выбрать для своего ПО удобный уровень гарантии разработки, чтобы облегчить работу? Конечно, нет. Уровень критичности в основном определяется процессом оценки безопасности, который предшествует применению DO-178, а также применимыми техническими стандартными приказами (Technical Standard Order, TSO) с учетом реальной эксплуатации. Почему? Потому что уровень гарантии связан с предполагаемым применением и проектом более крупной “системы”, частью которой является ПО. Например, вторичная навигация обычно имеет уровень C, но если тот же приемник всенаправленного азимутального радиомаяка/инструментальной системы посадки (VHF Omnidirectional Range/Instrument Landing System, VOR/ILS) используется при посадке по категории III с высотой принятия решения 0 футов, он может превысить минимальное требование TSO. Поэтому уровень не всегда определяется механически. FAA попыталась определить уровни в консультативном циркуляре (Advisory Circular, AC) 23.1309-1D для самолетов части 23 (14 CFR Part 23), а приложение A к этому циркуляру заходит туда, куда другие части правил не заходят. Но у каждой уникальной системы есть свой DAL, а ПО внутри этой системы имеет уровни критичности, равные DAL системы или ниже него. Ниже? Да, ПО или часть ПО может иметь уровень критичности ниже, чем система в целом. Более того, системы с ПО нескольких уровней критичности встречаются все чаще.

Почему бы тогда не сделать все ПО уровня A? Что ж, ПО уровня A, вероятно, качественнее ПО более низких уровней, так же как уровень B обычно качественнее уровня C. Так почему бы все не сделать уровнем A – самолет ведь станет безопаснее, верно? Возможно, но стоимость будет выше. В ПО, критичном для безопасности, бесплатных обедов не бывает. Больше безопасности стоит больше денег. Процесс оценки безопасности авионики определяет, какой объем безопасности требуется каждой системе, чтобы обеспечить приемлемый уровень безопасности. Предоставлять дополнительную безопасность просто не требуется. И без того достаточно трудно выполнить минимальное требование... Поднимать планку выше минимума, конечно, благородно, но может поставить компанию под финансовый риск. В данном случае благородство означает соблюдение минимального требования.

DO-178 содержит конкретные цели, зависящие от уровня критичности ПО. Более высокие DAL должны удовлетворять большему числу целей DO-178, чем более низкие уровни. После определения уровня критичности ПО вы обращаетесь к DO-178 и точно устанавливаете, какие цели должны быть удовлетворены для этого ПО. Теперь вы готовы к планированию. Здесь DO-178 похож на строительство дома: вы выполнили географический анализ, чтобы определить требуемый тип фундамента; это ваша “оценка безопасности”. Затем нужен процесс планирования, за которым следует процесс разработки. Параллельный процесс обеспечения корректности продолжается на протяжении планирования и разработки. Таким образом, разработка авиационного ПО по DO-178 похожа на строительство дома и следует той же трехфазной процессной логике.

DO-178, DO-254 и DO-278 имеют три интегральных процесса: планирование, разработку и интегральное обеспечение корректности:

Три интегральных процесса DO-178, DO-254 и DO-278

1. **Процесс планирования.** Выполняется первым.
2. **Процесс разработки.** Следует за планированием.
3. **Интегральные процессы.** Выполняются непрерывно на протяжении всего проекта.

Как видно на рисунке, сначала выполняется процесс планирования, а после его завершения начинается более крупный процесс разработки. На заднем плане постоянно идет самый большой процесс – обеспечение корректности, то есть интегральный процесс гарантии качества, управления конфигурацией (Configuration Management, CM), верификации и взаимодействия с сертифицирующим органом. Что подразумевается под планированием, разработкой и корректностью? Ниже дано краткое резюме.

Процесс планирования. До начала разработки или повторного использования ранее созданного либо унаследованного ПО нужно спланировать свои действия. При строительстве дома инспекторы сначала проверяют комплект планов, а затем регулярно инспектируют дом по мере строительства: фундамент, стены, электрику, сантехнику, крышу и готовое здание. DO-178 устроен аналогично. С процессом планирования связаны пять планов и три стандарта. Следующий блок суммирует ключевые аспекты этих пяти планов:

Пять планов процесса планирования

- Планы должны предшествовать работам по разработке.
- Необходимо доказать, что планы выполняются.
- Планы определяют, что, когда и кто выполняет, а также немного "как".
- Планы обычно утверждаются гарантией качества (Quality Assurance, QA), инженером по верификации соответствия (Compliance Verification Engineer, CVE); в США – представителем Федерального управления гражданской авиации США (Federal Aviation Administration, FAA) по инженерной экспертизе (Designated Engineering Representative, DER).

Гарантия качества является основным подписантом пяти планов. Почему? Потому что затем гарантия качества отвечает за аудит соблюдения этих планов инженерами. Если гарантия качества не рассмотрела и не утвердила планы, от нее нельзя разумно ожидать тщательных аудитов на соответствие этим планам. В Северной Америке гражданская авиация под управлением FAA исторически дополняет сертификационные действия назначением представителей FAA по инженерной экспертизе (Designated Engineering Representatives, DER), которые помогают в сертификации, а иногда могут даже формально получать полномочия на “одобрение” и фактически действовать как аудиторы или утверждающие лица от FAA. В Европе вместо DER иногда используется менее

полномочная роль – инженер по верификации соответствия (Compliance Verification Engineer, CVE), у которого меньше полномочий, чем у DER, но который все равно поддерживает гарантию качества. Пять планов, часто называемых “ключами DO-178”, кратко представлены ниже:

Пять планов DO-178

1. **План сертификации ПО** (Plan for Software Aspects of Certification, PSAC).
2. **План гарантии качества ПО** (Software Quality Assurance Plan, SQAP).
3. **План управления конфигурацией ПО** (Software Configuration Management Plan, SCMP).
4. **План разработки ПО** (Software Development Plan, SDP; в этом рисунке – SWDP).
5. **План верификации ПО** (Software Verification Plan, SVP; в этом рисунке – SWVP).

Пять планов – краткое изложение

1. **План сертификации ПО** (Plan for Software Aspects of Certification, PSAC): общий обзор того, как разработка ПО проекта будет соответствовать DO-178, включая ссылки на безопасность и систему, как описано в предыдущих главах по ARP4761A и ARP4754A:

План сертификации ПО (PSAC)

- Обычно 40-60 страниц; следует стремиться к краткости.
- Дает обзор системы, архитектуры и ПО.
- Рассматривает безопасность, критичность и сертификацию.
- Содержит ссылки на другие планы и артефакты.
- Готовится с участием представителя Федерального управления гражданской авиации США (Federal Aviation Administration, FAA) по инженерной экспертизе (Designated Engineering Representative, DER) и одобряется им.
- Представляется сертифицирующему органу и одобряется им.

1. **План гарантии качества ПО** (Software Quality Assurance Plan, SQAP): подробно описывает, как в данном проекте будут выполнены цели DO-178 по гарантии качества, включая одобрения, аудиты и соответствующее ведение записей:

План гарантии качества ПО (Software Quality Assurance Plan, SQAP)

- Рассматривает роль гарантии качества во всем процессе разработки.
- Обеспечивает согласованность планов, их включение в процесс и соблюдение.
- Обеспечивает соблюдение критериев перехода.
- Рассматривает проверки соответствия и инспекции.
- Содержит указания и графики аудитов и рассмотрений гарантии качества, включая контрольные перечни.

1. **План управления конфигурацией ПО (Software Configuration Management Plan, SCMP)**: подробно описывает, как в данном проекте будут выполняться цели DO-178 по управлению изменениями, базовым версиям и хранению:

План управления конфигурацией ПО (Software Configuration Management Plan, SCMP)

- Идентификация единиц данных.
- Использование системы управления конфигурацией.
- Критерии для управления конфигурацией разработки.
- Сообщения о проблемах и отслеживание изменений.
- Доступность данных и права доступа.
- Безопасность, сопровождаемость и повторяемость.
- Воспроизведение ПО, включая двоичный код, всех создаваемых данных и повторяемость для производства.

1. **План разработки ПО (Software Development Plan, SDP)**: суммирует, как будут выполняться разработка требований к ПО, проектирование, кодирование и интеграция с использованием связанных процессов, стандартов и инструментов для удовлетворения целей разработки DO-178:

План разработки ПО (Software Development Plan, SDP; в этом рисунке – SWDP)

- Определяет график, состав исполнителей, зависимости, поставляемые материалы, вехи и участвующие организации.
- Описывает среду разработки и используемые инструменты.
- Описывает процессы жизненного цикла: требования, проектирование, кодирование, интеграцию, неформальное тестирование при разработке, если оно применяется, и обработку изменений.
- Описывает операционную систему реального времени, OCPB (Real-Time Operating System, RTOS), обособление, ранее разработанное ПО и компоненты параметрических данных (Parameter Data Item, PDI).
- Описывает участие управления конфигурацией и гарантии качества, включая критерии перехода и поставляемые материалы.

1. **План верификации ПО (Software Verification Plan, SVP):** суммирует мероприятия по рассмотрению, тестированию и анализу, а также связанные инструменты верификации, необходимые для удовлетворения целей верификации DO-178:

План верификации ПО (Software Verification Plan, SVP; в этом рисунке – SWVP)

- Описывает рассмотрения и приемку результатов.
- Описывает трассируемость и верификацию.
- Охватывает тестирование и анализ.
- Включает среды тестирования и инструменты.
- Описывает регрессионное тестирование, повторную верификацию и квалификацию инструментов тестирования.
- Включает ресурсы и управление верификацией.

Типовые виды верификационных работ:

- функциональное тестирование;
- тестирование на робастность;
- тестирование штатного диапазона;
- анализ структурного покрытия.

Три стандарта

DO-178, а также DO-254 и DO-278, признает, что требования, проектирование и реализация (код) принципиально важны и должны оцениваться. Для оценки нужно сравнить требования, проект ПО и код с заранее определенными критериями. Однако такие критерии по своей природе субъективны. В DO-178C есть 71 “цель”, и они объективны, потому что каждую из них можно однозначно и полно оценить. Но требования, проект и код конкретной системы могут иметь очень разные критерии

оценки. По этой причине DO-178 не задает явных целей для требований, проектирования или кода. Вместо этого каждый проект определяет собственные явные критерии в трех подробных стандартах:

Стандарт на требования к ПО: задает критерии декомпозиции и оценки системных требований в требования высокого уровня к ПО (High-Level Requirements, HLR), а также критерии декомпозиции требований высокого уровня в требования низкого уровня (Low-Level Requirements, LLR), включая производные требования и требования, связанные с безопасностью. Критерии работы с требованиями низкого уровня проект может описывать либо в этом стандарте, либо в стандарте проектирования ПО: в DO-178С граница между требованиями низкого уровня и проектом задается самим проектом.

Стандарт проектирования ПО: задает критерии уточнения требований низкого уровня до состояния, пригодного для реализации, а также критерии определения и оценки архитектуры ПО и проекта ПО, включая внутренние и внешние интерфейсы, поток данных и поток управления с анализом межкомпонентной связности.

Стандарт кодирования ПО: задает критерии реализации и оценки исходного кода ПО. Обычно они выражаются как подмножество руководства MISRA-C, подготовленного Ассоциацией по надежности ПО в автомобильной промышленности (Motor Industry Software Reliability Association, MISRA).

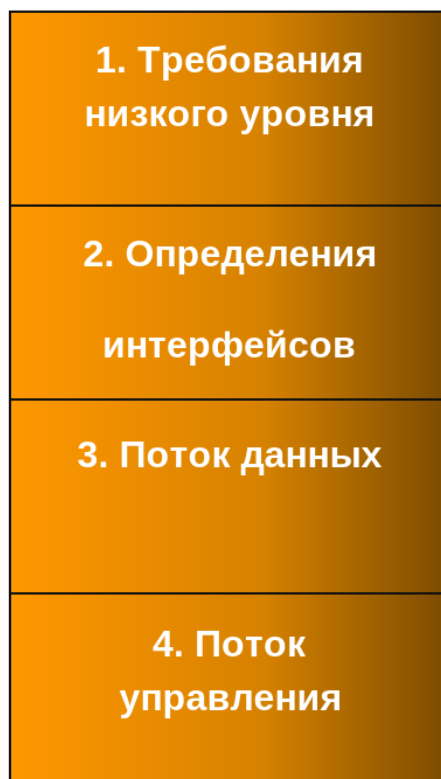
Эти пять планов и три стандарта составляют документы планирования, требуемые для DO-178. Поскольку уровень D является наименее строгим из пяти уровней DO-178, три стандарта для него не требуются; они требуются для уровней А-С. Уровень Е практически не является уровнем DO-178, поскольку для самого низкого уровня нет обязательных целей DO-178. Нужна лишь оценка функциональных опасностей (FHA), доказывающая, что это уровень Е и, следовательно, функция не связана с безопасностью.

Таким образом, три стандарта – требования, проектирование и код – кодифицируют проектно-специфические критерии. Требования подробнее рассматриваются в одной из следующих глав этой книги. Стандарты кодирования обычно просто переиспользуются из других отраслевых стандартов кодирования, например стандартов Института инженеров электротехники и электроники (Institute of Electrical and Electronics Engineers, IEEE) или руководств MISRA для кода на С. Но проект ПО по DO-178 уникален: он охватывает проектно-специфические приемы, процессы и критерии документирования проекта ПО, чтобы затем можно было выполнить его верификацию. Однако где тонкие детали требований к ПО начинают пересекаться с архитектурой ПО, и можно ли полностью знать все детали требований до начала разработки этой архитектуры? Ответы связаны с тем, как DO-178С рассматривает отношения между требованиями, проектом и кодом, как показано ниже:

Взаимосвязь требований, проектирования и реализации в DO-178C:



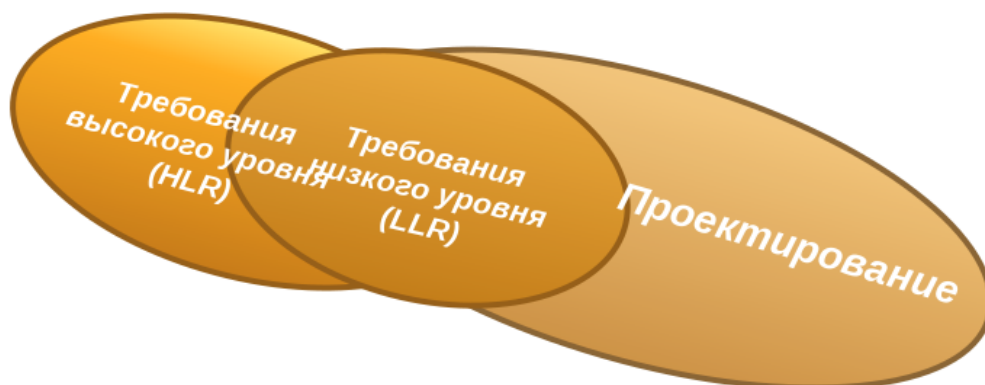
Ключевые аспекты стандарта проектирования ПО по DO-178C суммированы на следующем рисунке:



Ключевые аспекты проектирования ПО по DO-178C

Поскольку проект ПО является мостом между требованиями к ПО и кодом, спецификация проекта и ее верификация относятся к ключевым характеристикам разработки ПО по DO-178C. Важно понимать, что граница между требованиями высокого уровня, требованиями низкого уровня и проектом “гибкая”: каждый проект просто

определяет эти границы в стандарте на требования к ПО и стандарте проектирования ПО. По существу, требования низкого уровня образуют область перекрытия требований высокого уровня и проекта, как показано ниже:



Требования низкого уровня в DO-178C: перекрытие требований высокого уровня и проектирования

Несколько слов о потоке данных и потоке управления.

Проект ПО по DO-178C для DAL C-A должен включать документирование, а затем анализ потоков данных и управления ПО. Это делается для обеспечения детерминизма и оценки связности по данным и связности по управлению. Связность по данным суммирована на следующем рисунке:

Связность по данным

Связность по данным – это зависимость программного компонента от данных, не находящихся под исключительным контролем этого компонента.

Связность по данным предполагает все глобальные переменные, а также те локальные переменные, которые передаются вниз через списки параметров компонентам нижнего уровня, например функциям или процедурам.

Связность по управлению суммирована на следующем рисунке:

Связность по управлению

Связность по управлению – это способ или степень, в которой один программный компонент влияет на выполнение другого программного компонента.

Связность по управлению возникает, когда один модуль передает данные, используемые для управления потоком выполнения другого модуля.

Поскольку и связность по данным, и связность по управлению можно оценивать по документации проекта ПО DO-178C, анализ межкомпонентной связности выполняется по четырем основным причинам, указанным ниже:

Четыре основные роли анализа связности в DO-178

1. Определить интерфейсы модулей и зависимости по данным между компонентами.
2. Подтвердить завершение интеграционного тестирования и детерминизм.
3. Определить глубину интерфейса.
4. Проанализировать и протестировать указанное выше; изменить проект или код, чтобы снизить межкомпонентную связность и усилить внутреннюю связность.

Четыре основные роли анализа межкомпонентной связности в DO-178

После того как проектно-специфические планы и стандарты DO-178 рассмотрены и утверждены отделом гарантии качества и соответствующим сертифицирующим органом, можно переходить к формальным работам по разработке ПО. Конечно, если вы уже выполняли проекты по DO-178 или проекты ПО, критичного для безопасности, у вас, скорее всего, есть базовые планы и стандарты программной инженерии, которые можно переиспользовать при подготовке проектно-специфических планов и стандартов DO-178. На практике предварительные требования к ПО и проект ПО часто начинают разрабатывать параллельно с планами и стандартами: это экономит время и помогает уточнить детали самих планов и стандартов. Однако теоретически DO-178 предполагает классическую каскадную модель с последовательными процессами планирования, разработки и верификации. Почему DO-178 в целом следует такому каскадному подходу?

Потому что так сложилось исторически...

Как отмечалось в предыдущей главе, DO-178 впервые был составлен в 1980-х годах, когда вместе с усложнением систем формировалась программная “инженерия”. Дважды пересмотренный к началу 1990-х, DO-178 в целом отражал ряд лучших практик и философий разработки ПО, распространенных в то время, включая модель зрелости возможностей (Capability Maturity Model, CMM), военные стандарты и ныне печально известную каскадную модель (Waterfall model). Сегодня почти никто не использует чистую каскадную модель. На практике ее сторонники часто применяют разработку на основе моделей (Model-Based Development, MBD), особенно совместно с DO-178C и DO-331. Типовая V-модель и даже безопасное подмножество Agile также часто применяются, причем Agile чаще встречается на более низких DAL.

Хотя DO-178 допускает практически любую детерминированную и верифицируемую методологию разработки ПО, следы каскадной модели прочно засели в умах многих практиков и аудиторов в области разработки авионики, особенно у людей с сидящими волосами, вроде автора. Какие признаки каскадного подхода обычно используются?

- сопровождение V-моделью с оценкой выходных данных каждой фазы;
- обильная документация;
- последовательное прохождение и переходы между требованиями, проектом, кодом, интеграцией и тестированием.

Теперь вернемся к планированию разработки авиационного ПО. Вы уже выполнили полноценную оценку функциональных опасностей (Functional Hazard Assessment, FHA) на основе ARP4761. У вас есть добротные системные требования на основе ARP4754A, учитывающие требования безопасности. У вас есть отличные планы и стандарты, полностью соответствующие DO-178. Великолепно: кажется, можно начинать разработку ПО и писать код. Срочно собирайте прототип, чтобы показать руководству и заказчику, какие вы профессионалы! Но есть одна проблема: вы еще даже близко не готовы писать настоящее авиационное ПО. Почему? Сейчас разберем.

Причины, по которым вы еще не готовы писать ПО:

Причины, по которым вы еще не готовы писать ПО

1. Нет формальных требований к ПО, поэтому неизвестно, что создается, и невозможно согласовать результат.
2. Нет формальной архитектуры ПО, основанной на требованиях к ПО.
3. Нет завершенного проекта ПО, трассируемого к требованиям высокого уровня к ПО.
4. Пока нет утвержденных стандартов кодирования.
5. Нет контрольных перечней рассмотрения, поэтому неизвестны критерии приемки требований, проекта, кода и интеграции.
6. Пока нет контрольных перечней аудита гарантии качества ПО для аудита мероприятий разработки ПО.
7. Может отсутствовать система управления трассируемостью для текущих мероприятий по трассируемости.
8. Может отсутствовать работоспособный подход к тестированию ПО и соответствующие инструменты для быстрой верификации ПО.

Что, нельзя просто придумать требования к красивому дому, купить стройматериалы и позвать друзей строить дом мечты под приятные напитки? Можно, но не в том случае, если вы когда-нибудь захотите законно жить в этом доме или продать его. С авионикой то же самое: можно долго спорить, является ли DO-178 формально “обязательным”, но никто всерьез не спорит, что для законной продажи авионических систем по всему миру ПО

должно соответствовать DO-178. Что из этого следует? Все просто: каждую из причин, перечисленных выше, нужно устранить ДО того, как вы начнете писать ПО. Почему? Для этого есть и нормативные, и деловые причины.

С нормативной точки зрения важно иметь возможность оценивать каждый артефакт в процессе разработки. У каждого шага разработки есть входные и выходные критерии, а они связаны с конкретными критериями приемки. Где определены такие критерии приемки? В ваших стандартах разработки – помните стандарт на требования, стандарт проектирования и стандарт кодирования? – и в связанных контрольных перечнях для каждого артефакта. Контрольные перечни? Да: авиация основана на контрольных перечнях. Каждый раз перед взлетом пилот и экипаж выполняют проверки по контрольным перечням. Когда коммерческий самолет, почти всегда с пилотом и вторым пилотом, готовится к посадке, происходит то же самое. Пилот может сказать: “Подготовка к посадке; закрылки 10 градусов”, после чего второй пилот подтверждает, проверяя положение закрылков и объявляя: “Подтверждаю. Закрылки 10 градусов”. Что произошло? Пилот следовал письменному контрольному перечню посадки, а второй пилот проверял интерпретацию и действия пилота по этому перечню. Тем самым второй пилот выполнил “независимое рассмотрение” обязательных действий пилота. Разве пилот и второй пилот не выучили посадочные контрольные перечни наизусть и им не нужно фактически следовать письменному перечню в кабине? Опытные пилоты действительно должны знать его на память. Но, как всем известно, память может подвести, особенно под давлением, а контрольные перечни заранее записываются и формально используются при реальном выполнении работы. Бумажные они или цифровые – неважно. Важно, что для обязательных действий определены критерии, указанные в контрольном перечне. Каждый контрольный перечень рассмотрен, поставлен под управление конфигурацией и утвержден. Это означает, что для каждой операции существует одна и только одна действующая версия каждого контрольного перечня. С DO-178 все так же.

Краткое примечание о целях, критериях перехода и контрольных перечнях. Цели DO-178 указаны в приложениях в конце официального документа DO-178. Критерии перехода действуют как критерии приемки, соотнесенные с этими целями. Переходы связаны с моделью жизненного цикла, а не со стандартами. Контрольные перечни полезны, но их нужно соотносить с критериями перехода жизненного цикла. Они находятся на один уровень ниже.

DO-178 использует контрольные перечни, которые адаптируются под специфику проекта и стандарты. Эти перечни также кодифицируют критерии успешности, применимые к соответствию DO-178. Для каждого основного артефакта контрольные перечни подтверждают следующее:

- заранее формально определены критерии приемки для каждого процесса и артефакта;
- при рассмотрении учтены как минимум критерии из контрольного перечня;
- можно доказать факт рассмотрения и аудита, включая независимость там, где она требуется.

Что рассматривается по контрольным перечням? Практически все, особенно элементы, связанные с обязательными целями DO-178: планы, стандарты, безопасность, требования, проект, код, тестирование и результаты, квалификация инструментов, поставщики и так далее.

Когда планы, стандарты и контрольные перечни DO-178 готовы, можно начинать разработку ПО. К этому моменту вы должны пройти первый из четырех обязательных сертификационных этапов взаимодействия с сертифицирующим органом (Stage Of Involvement, SOI), поскольку именно SOI 1 оценивает эти планы, стандарты и контрольные перечни. В классическом каскадном подходе следующие действия обычно выполняются последовательно, чтобы показать выполнение входных и выходных критериев, которые в терминологии DO-178 называются “переходами”.

Классические действия по разработке ПО по DO-178

Классические действия по разработке ПО по DO-178

1. Завершить требования системного уровня.
2. Распределить требования системного уровня на ПО, аппаратуру или оба компонента.
3. Декомпонировать системные требования в требования высокого уровня к ПО, включая при необходимости производные требования.
4. Декомпонировать требования высокого уровня к ПО в требования низкого уровня, включая производные требования.
5. Разработать проект ПО, включая уточнение требований низкого уровня до уровня детализации, достаточного для реализации.
6. Реализовать исходный код.
7. Интегрировать ПО, включая продукты третьих сторон.

Разумеется, каждое из перечисленных выше действий по разработке ПО будет выполняться в соответствии со следующими ранее рассмотренными и принятыми документами:

1. План разработки ПО.
2. Стандарты на требования к ПО, проектирование ПО и кодирование ПО.
3. Соответствующий контрольный перечень для каждого действия, рассматриваемый независимо для применимых действий разработки ПО уровней А и В.

На практике современные сложные системы, критичные для безопасности, включая авионику, редко следуют такой строгой каскадной последовательности. Чаще встречаются разработка на основе моделей, бережливые методы, прототипирование и меняющиеся требования заказчика. В этом и красота DO-178: он не является предписывающим, то есть допускает гибкость. Но какой бы метод разработки ПО вы ни выбрали, включая повторное

использование унаследованного ПО, этот метод нужно заранее подробно описать в рассмотренных планах, стандартах и контрольных перечнях. Затем придется доказать, что вы действительно им следовали.

Когда реализация ПО завершена, у вас есть требования, трассируемые к коду, код трассируется обратно к требованиям, а требования, проект и код задокументированы и рассмотрены. Все готово, верно? Едва ли. Вы всего лишь готовы к следующему этапу взаимодействия с сертифицирующим органом, то есть SOI 2. Что делает SOI 2? Он подтверждает, что ваш проект соответствует планам и стандартам, а это можно доказать через контрольные перечни рассмотрений.

SOI 2 также должен подтвердить, что собственные проектные аудиты управления конфигурацией и гарантии качества ПО (Software Quality Assurance, SQA) проводились на протяжении реализации. Что такое аудит SQA? DO-178 – это предварительное планирование, а затем доказательство, обычно через контрольные перечни, что работа выполнена по планам и стандартам. Однако чем выше уровень критичности, тем меньше допускается единоличное доверие к автору, потому что на кону человеческие жизни. У FAA есть письменный приказ (FAA Order 8110-49, Chapter 3), который задает уровень участия FAA (Level of FAA Involvement, LOFI) на основе критичности и учитывает множество других факторов, например опыт разработчика в сертификации ПО. Чем выше критичность, тем выше риск в глазах сертифицирующих органов. Поскольку их ресурсы ограничены, они обычно сосредотачиваются на проектах с более высоким риском, особенно если их выполняют новички. Фактически в DO-178 есть несколько уровней лиц, участвующих в рассмотрениях или аудитах:

1. Техническое рассмотрение выполняется для завершенного артефакта, “готового к рассмотрению”, чтобы оценить его соответствие планам, стандартам и контрольному перечню. Более высокие уровни критичности требуют независимости рассмотрения, например участия технического специалиста, который не был автором артефакта.
2. Аудиты SQA выполняются для оценки соблюдения процесса, а не обязательно технического соответствия. Использовались ли при создании артефакта процессы, соответствующие планам и стандартам? Соблюдались ли критерии перехода, то есть корректное использование входных и выходных критериев каждого процесса?
3. Аудиты взаимодействия с сертифицирующим органом, например аудиты представителей FAA по инженерной экспертизе (DER) в США, оценивают предыдущие технические рассмотрения и аудиты SQA.
4. Сертифицирующий орган, например Агентство Европейского союза по авиационной безопасности (European Union Aviation Safety Agency, EASA) или FAA, проверяет все перечисленное выше.

Как видно, SQA в авионике отличается от SQA в других отраслях. Там SQA часто выполняет технические рассмотрения и тестирование, а DO-178 сосредотачивает SQA на двух основных действиях: сначала определить или одобрить проектные планы и

стандарты, а затем с помощью аудитов оценить, соблюдались ли эти планы и стандарты надлежащим образом. DO-178 не диктует, как именно SQA должна выполнять работу. Он просто требует, чтобы SQA достигала минимального набора целей и сохраняла эту позицию на протяжении процесса. Просто. В теории, разумеется.

После утверждения SOI 2 реализация ПО в основном завершена, хотя всегда остаются небольшие изменения требований и исправления дефектов. Но еще нужно выполнить полное тестирование ПО. В других областях тестирование ПО часто считают “искусством”. Проблема искусства в том, что оно субъективно: мы не можем договориться об определении искусства, но “узнаем его, когда видим”. Беда в том, что люди никогда полностью не соглашаются, что является хорошим искусством, а что плохим. То же самое с музыкой, едой, погодой и так далее: все это субъективно. Поэтому DO-178 превращает искусство тестирования ПО в науку, требуя подробного тестирования как требований к ПО, так и кода. Специалисты по тестированию ПО по DO-178 должны уметь составлять тестовые примеры, чтобы оценить, соответствует ли логика всему объему письменных требований. DO-178 также вводит понятие “мы уже закончили?”. Если стремиться к совершенству, тестирование ПО могло бы охватить практически бесконечное число комбинаций и перестановок. DO-178 ограничивает это даже для самого высокого уровня критичности. Кроме того, чем выше уровень критичности ПО, тем в большей степени нужно показать, что код этого ПО покрыт тестами, то есть выполнить анализ структурного покрытия.

Классические действия по верификации ПО по DO-178

Слышали ли вы когда-нибудь фразу: “Я инженер по верификации и валидации (Verification and Validation, V&V); некоторые считают меня экспертом”? Тестирование ПО всегда отчасти искусство, а отчасти наука. В DO-178 специалисты по тестированию ПО действительно должны хорошо понимать проект ПО и код. В неавиационных областях “тестировщики ПО” часто мало знают о тонкостях проекта ПО и кода. В DO-178 это не так. Сначала немного о V&V. Как всем известно, V&V – распространенное обозначение верификации и валидации. К сожалению, реальный смысл V&V, особенно в авиационном ПО, часто понимается неправильно. О V&V ПО написаны сотни книг, поэтому следующий абзац, конечно, не может полноценно раскрыть эту весьма объемную тему.

Верификация – это оценка результата процесса, чтобы определить, соответствует ли выходной результат спецификации. Что тогда означает “верификация” для авиационного ПО? Технические специалисты лучше всего работают с количественными уравнениями; если вы не технический специалист, то зачем вы это читаете? Итак, верификация в авионике выражается следующим уравнением:

$$V = R + T + A$$

Где:

V – это верификация

R – это *рассмотрение*

T – это *тестирование*

A – это *анализ*

(Обратите внимание, что “А” для анализа намеренно показана как малая добавка: анализ используется только тогда, когда сочетание рассмотрения и тестирования не полностью верифицирует требование.)

Что это уравнение означает для авиационного ПО? Ответ такой:

Артефакты, созданные человеком, должны рассматриваться по контрольным перечням.

Требования к ПО и код должны тестироваться.

Если тестирование само по себе не дает окончательного вывода, требуется дополнительный анализ.

Разумеется, объем рассмотрений, тестирования и анализа полностью зависит от уровня критичности ПО, как будет показано ниже. Это кратко описывает верификацию. А что насчет валидации? Помните: верификация оценивает артефакт, чтобы определить, соответствует ли он спецификации. Для верификации ПО спецификацией являются требования. Но что делать, если сами требования не очень хороши? Здесь появляется валидация.

Валидация оценивает требования, чтобы определить, являются ли они хорошими: ясными, краткими, корректными, полными и верифицируемыми. Мусор на входе – мусор на выходе. Самый важный вход для качества авиационного ПО – требования. Значит ли это, что требования к авиационному ПО должны валидироваться? На самом деле DO-178 этого не требует, потому что валидация требований к ПО без завершенной системы считается субъективной, а DO-178 сосредоточен на объективных критериях. Официально требования должны валидироваться на уровнях аппаратуры и системы, соответственно по DO-254 и ARP-4754A. Неофициально лучшие практики должны включать валидацию требований к ПО в ходе их рассмотрения, например проверку корректности и полноты.

Как упоминалось выше, тестирование ПО – отчасти искусство и отчасти наука. И, как и при разработке ПО, совершенство невозможно. Однако DO-178 был написан отраслевыми специалистами при лишь небольшом государственном надзоре, поэтому акцент сделан на разумной экономической эффективности. На тестирование ПО легко потратить больше ресурсов, чем на его разработку. За весь жизненный цикл авиационного изделия на тестирование наверняка будет потрачено больше времени, чем на разработку.

Из чего состоит разумное и экономически эффективное тестирование ПО? Во-первых, из наличия хороших требований к ПО и их полной верификации. Далее нужно дополнительное тестирование на робастность, которое пытается найти ошибки в логике через стрессовое тестирование и тестирование производительности, проверку граничных и недопустимых значений, а также выполнение всех переходов. Наконец, нужен анализ

структурного покрытия, чтобы убедиться, что вся логика, которая потенциально может выполняться на самолете, верифицирована и ведет себя как задумано. Трассируемость сверху вниз формализуется, чтобы доказать верификацию всех документированных требований. Трассируемость снизу вверх включается, чтобы доказать отсутствие недокументированной логики. Будьте внимательны: трассируемость снизу вверх – это не просто инверсия матриц. Это означает, что каждый сегмент кода должен трассироваться вверх к требованию: высокого уровня, низкого уровня или производному. Глубина этого сегмента кода соотносится с глубиной, требуемой уровнем критичности.

Каков масштаб и относительный объем работ при тестировании по DO-178? Возможно, поможет рисунок:

Относительный объем работ по тестированию ПО по DO-178 – оптимум (уровень А)

- **Тестирование требований к ПО.**
 - Требования высокого уровня.
 - Требования низкого уровня.
- **Тестирование робастности и производительности.**
 - Граничные значения.
 - Ошибочные условия.
- **Структурное покрытие ПО.**
 - Операторы.
 - Решения и условия.
 - Модифицированное покрытие условий/решений (Modified Condition/Decision Coverage, MC/DC).

Относительный объем работ по тестированию ПО по DO-178 – оптимум (уровень А)

Первый приоритет: тесты, построенные по требованиям, например функциональные тесты.

В DO-178 тестирование прежде всего строится от требований. В новой версии, DO-178C, этот принцип усилен: каждый значимый фрагмент кода должен трассироваться как минимум к одному требованию. DO-178 не задает формального порога детализации требований, поэтому качество тестов напрямую зависит от качества самих требований. Чтобы компенсировать риск слишком поверхностных требований, для уровней А-С одних номинальных функциональных тестов недостаточно: в набор тестов, основанных на требованиях, должны входить также проверки робастности, а результаты тестирования затем дополняются анализом структурного покрытия.

Если требования написаны хорошо, тесты по ним обычно уже закрывают большую часть последующей работы: около 90 % необходимых сценариев робастности и около 80 % кода для уровня В. Почему так? Хорошие требования достаточно подробно описывают низкоуровневую функциональность, границы, недопустимые значения, ошибочные условия и ожидаемое поведение системы в таких случаях. Эти условия можно вывести из требований, поэтому 80-90 % необходимых условий покрываются уже при анализе требований и подготовке тестовых случаев по ним.

Если же требования слабые, тесты, написанные по таким требованиям, могут закрыть только 50-60 % нужного покрытия. Недостающие детали всплывут позднее, во время анализа структурного покрытия: вы увидите код, который не привязан к достаточным требованиям, и придется возвращаться назад, дописывать требования и переделывать тесты. Для уровней D и E анализ структурного покрытия не требуется, поэтому там такую проблему можно обнаружить еще хуже. Как обычно, дешевле сделать нормально сразу, чем потом героически чинить собственную экономию. Вывод простой: вкладывайтесь в хорошие требования заранее и подкрепляйте их структурированными рассмотрениями и контрольными перечнями.

Важно отметить, что пять уровней критичности DO-178 требуют существенно большего объема тестирования ПО по мере роста критичности. На разработку ПО уровень критичности влияет меньше, но для тестирования это не так, что и показано на следующем рисунке, суммирующем тестирование, требуемое для каждого DAL.

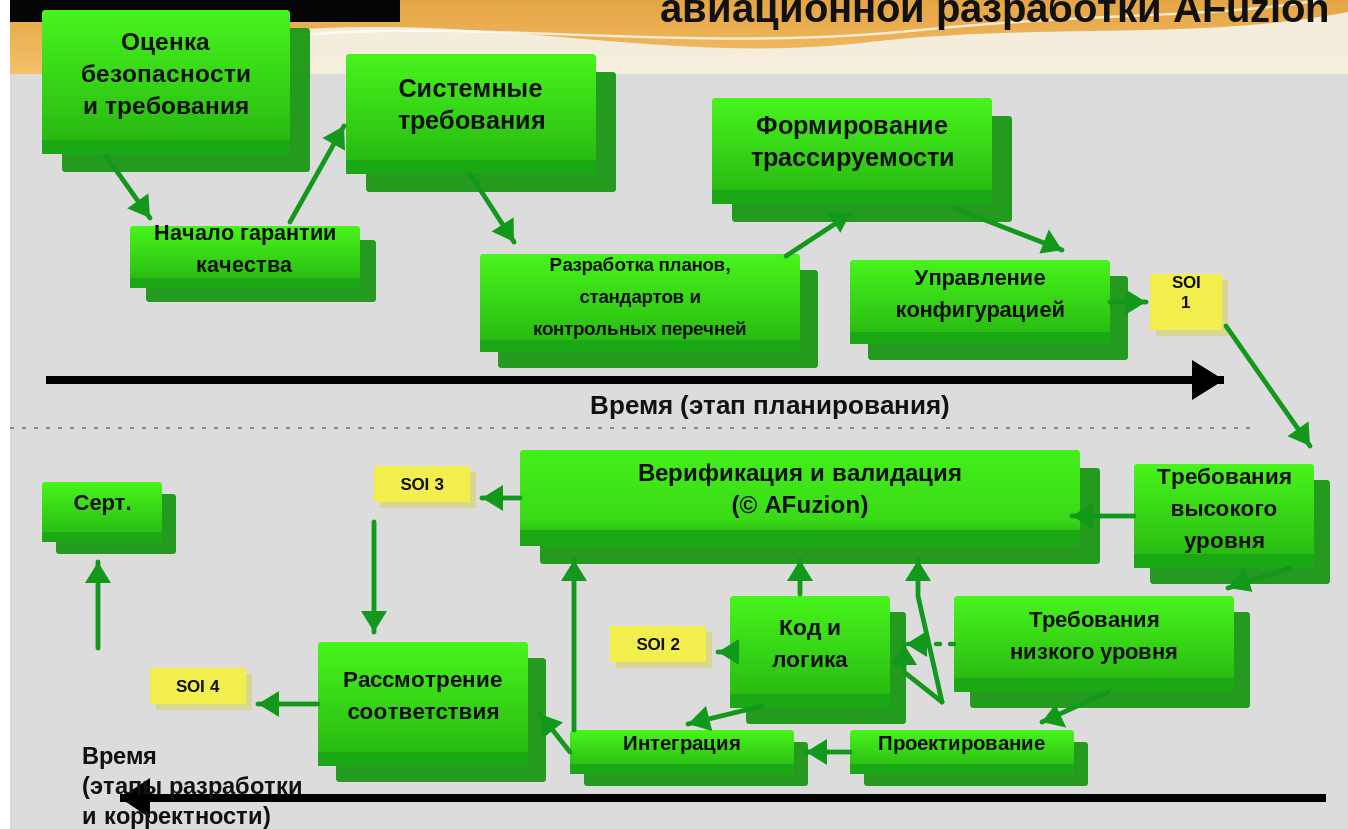
Основные различия в тестировании между уровнями критичности DO-178:

Основные различия в тестировании между уровнями критичности DO-178

- **Уровень Е.** Тестирование ПО не требуется.
- **Уровень D.**
 - Рассмотрение требований высокого и низкого уровня.
 - Тестирование требований высокого уровня.
- **Уровень C.** То же, что для уровня D, плюс:
 - Рассмотрение проекта и кода.
 - Тестирование требований низкого уровня.
 - Верификация робастности и покрытия каждого оператора исходного кода.
 - Верификация связности по данным и управлению.
- **Уровень B.** То же, что для уровня C, плюс:
 - Тестирование целевой системы.
 - Доказательство покрытия решений/условий кода.
 - Независимость мероприятий покрытия кода.
- **Уровень A.** То же, что для уровня B, плюс:
 - Доказательство модифицированного покрытия условий/решений.
 - Доказательство корреляции исходного и двоичного кода.

После тестирования SOI 3 оценивает полноту действий по верификации. Затем SOI 4 сопровождает рассмотрение соответствия, которое охватывает соблюдение всех применимых целей DO-178, включая промежуточные изменения. Теперь соберем все вместе.

Следующий рисунок, показывающий оптимальную реализацию DO-178, которую автор использует в большинстве учебных курсов, называется “Зеленая змея” (Green Snake):



Оптимальный маршрут реализации DO-178C

Как показано выше, “Зеленая змея” начинается в верхнем левом углу с процесса безопасности, затем идет слева направо по мере выполнения системного и программного планирования и завершается рассмотрением планов и стандартов на SOI 1 до официального начала разработки. Затем в нижней половине справа налево выполняются разработка, тестирование и рассмотрение соответствия. Разумеется, “сертификация” происходит только в контексте системы и воздушного судна, при этом SOI 4 обычно должен быть завершен до летных испытаний, чтобы все аспекты, связанные с безопасностью, были подтверждены до этих испытаний. Почему змея зеленая? Все просто: по мнению автора, хотя DO-178 гибок и не является предписывающим, описанный выше путь “Зеленой змеи” оптимально снижает общую стоимость. Автор американец, а американские деньги зеленые: хотите сэкономить – следуйте “Зеленой змее”.

Аппаратура авионики по DO-254

Автор: Вэнс Хилдерман

Рецензент:

Г-н Франсуа Гуэй, представитель Федерального управления гражданской авиации США (Federal Aviation Administration, FAA) по инженерной экспертизе (Designated Engineering Representative, DER)

DO-254 иногда называют “младшим родственником DO-178”. Но, как и со многими младшими братьями и сестрами, слово “младший” часто вводит в заблуждение, а в некоторых вопросах сомнительна и сама аналогия. DO-254, или точнее “Руководство по гарантии разработки бортовой электронной аппаратуры” (*Design Assurance Guidance for Airborne Electronic Hardware*), появился как очевидный ответ на три одновременных и связанных между собой явления:

1. Аппаратная логика (firmware, в историческом употреблении автора) стала играть в авионике все более заметную роль, поскольку разработчики быстро осваивали решения на базе интегральных схем.
2. Процесс разработки такой аппаратной логики для авионики не регулировался, а сертификация выполнялась постфактум на системном уровне, как для “черного ящика”.
3. Появлялись свидетельства, что некоторые изготовители намеренно переносили логику из “ПО” в аппаратную реализацию в интегральной схеме, которую тогда часто называли термином firmware, чтобы избежать требований DO-178B, применимых к ПО.

В 1980-е и 1990-е годы инженерия встроенного авиационного ПО сильно продвинулась вперед, а разработка аппаратной логики для интегральных схем все еще воспринималась как неформальное вспомогательное ремесло. Но что автор называет термином firmware? Где проходит граница между soft, firm и hard – между ПО, конфигурируемой аппаратной логикой и фиксированной аппаратурой?

Краткая история DO-254

Два десятилетия назад аппаратная логика, реализуемая в интегральных схемах, в авионике в основном ограничивалась специализированными функциями. Сегодня ее роль гораздо разнообразнее и быстро расширяется. Раннее применение такой логики в авиации было ограничено по нескольким причинам:

1. Средства разработки аппаратной логики давали меньше гибкости по сравнению со средствами разработки ПО.
2. Такую логику было трудно изменить после прожига или загрузки в интегральную схему. Более удобные в применении ПЛИС (Field-Programmable Gate Array, FPGA) вышли на первый план среди программируемых устройств только к середине 1990-х годов, а более консервативная авиационная отрасль внедряла их медленнее.
3. Аппаратная логика считалась менее предпочтительным вариантом, чем ПО, из-за, как казалось, более сложной отладки и обновления – во многом по причинам из пунктов 1 и 2.

В каждой из этих причин была доля истины, но сильные инженеры и более развитые среды разработки обеспечили развитие и внедрение аппаратной логики в авионике. Особенно важны были успехи в области ПЛИС: они вывели такую логику на передний план в авиации. С появлением ПЛИС все перечисленные ограничения существенно ослабли. ПЛИС все активнее использовали современные средства разработки, легко обновлялись и давали возможные преимущества по гибкости и скорости выполнения по сравнению с логикой, реализованной в ПО.

Сертифицирующие органы, как гражданские, так и военные, требуют уверенности в том, что авиационная аппаратура разрабатывается с помощью последовательных и детерминированных процессов, в целом повторяющих жизненный цикл разработки ПО с учетом ключевых различий, описанных в этой главе. Традиционно аппаратуру можно было исчерпывающе испытать, поскольку она считалась “простой”. Для такой простой аппаратуры все комбинации входов и выходов можно было разумно проверить, поэтому аппаратуру испытывали именно так в составе системных испытаний. Но аппаратная логика, реализуемая в интегральных схемах, может быть столь же сложной, а часто и более сложной, чем ПО. Все комбинации входов и выходов для нее уже нельзя разумно испытать, поэтому такая аппаратура считается “сложной”, а не “простой”. Подробнее о сложной и простой аппаратуре будет сказано ниже.

DO-254 требует, чтобы процесс разработки аппаратуры включал требования к аппаратуре, концептуальное проектирование, детальное проектирование, реализацию, валидацию и верификацию, а затем переход к производству. На протяжении этих процессов фоном выполняются управление конфигурацией и гарантия процесса, близкая к гарантии качества ПО. Разработка аппаратуры уровней гарантии разработки (Development Assurance Level, DAL) C и D выполняется в основном как “черный ящик”: требуются прежде всего требования, испытания требований, трассируемость, а также

рассмотрения требований и испытаний, включая гарантию процесса. Для аппаратуры уровней А и В добавляется подход “белого ящика”: стандарты, рассмотрения и испытания фактического проекта и реализации логики.

Традиционно разработка аппаратуры отличалась от разработки ПО следующим образом:

- Сложность ПО росла быстрее, чем сложность аппаратуры.
- Команды разработки аппаратуры были более опытными и более компактными, чем команды разработки ПО.
- Значительную часть аппаратуры можно было постепенно прототипировать, параллельно выполняя испытания.
- Многие аспекты разработки аппаратуры можно было инспектировать или исчерпывающе испытывать.

Но с появлением современной технологии специализированных интегральных схем (Application-Specific Integrated Circuit, ASIC), а затем ПЛИС эти различия между ПО и аппаратурой стали исчезать. Поэтому фокус DO-254 сместился к сложной электронной аппаратуре (Complex Electronic Hardware, CEH), например к аппаратной логике, реализованной в ПЛИС, ASIC или других интегральных схемах. Меморандум 27 Группы сертификационных органов по ПО (Certification Authorities Software Team, CAST), а затем документ “Приемлемые методы определения соответствия 20-152A” (Acceptable Means of Compliance 20-152A, AMC 20-152A) дополнительно уточнили трактовку и практическое применение DO-254.

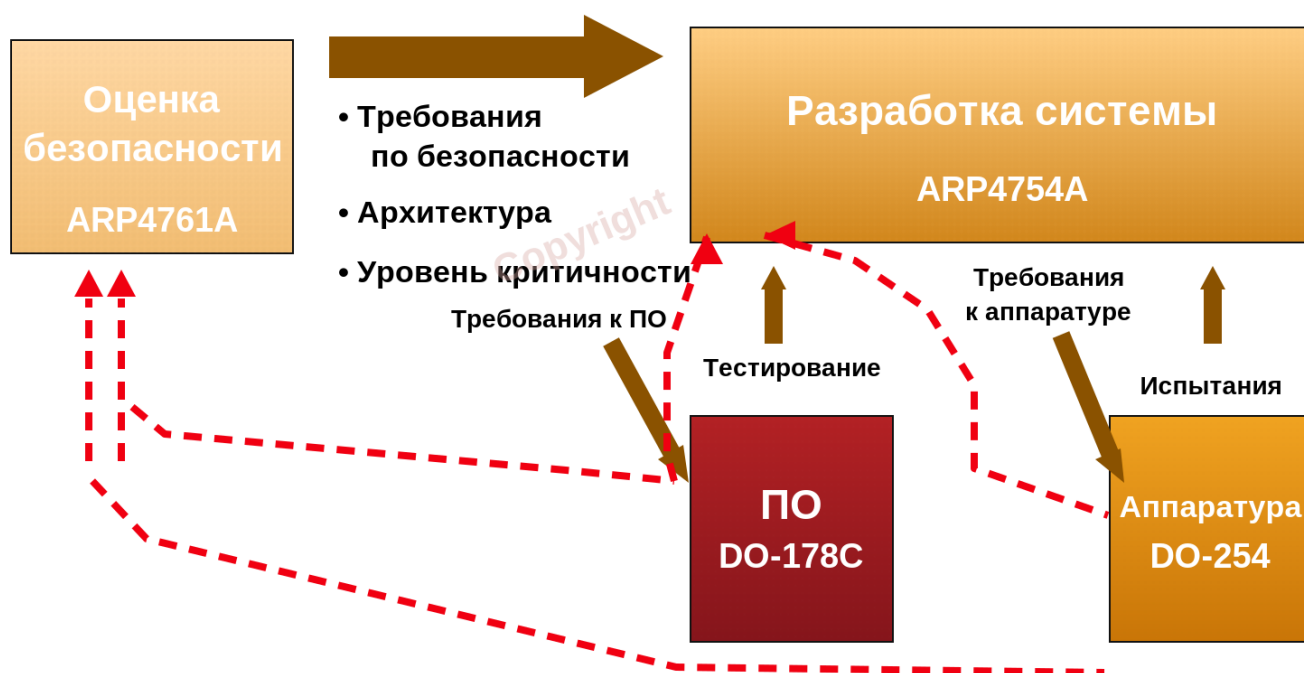
В результате этой эволюции, почти революции, аппаратной логики для интегральных схем разработчики авионики все чаще выбирали реализацию логики в аппаратуре, а не в ПО. Однако DO-178 строго не распространялся на аппаратную логику, реализованную в интегральных схемах, и нормативного аналога для нее не существовало. Поэтому такая аппаратная логика потенциально могла уйти от сертификационной строгости, связанной с ПО, хотя назначение логики в программной и аппаратной реализации могло быть почти одинаковым. Так возникла необходимость в DO-254: обеспечить, чтобы аппаратура и логика, разрабатываемые специально для авионики, проходили сопоставимый режим разработки и сертификации, как если бы они были реализованы в ПО по DO-178.

DO-254 и его европейский аналог ED-80 изначально предназначались для охвата всей электронной аппаратуры, включая линейно заменяемые блоки (Line Replaceable Unit, LRU), сборки печатных плат (Circuit Card Assembly, CCA), готовые коммерческие изделия (Commercial Off-The-Shelf, COTS), относящиеся к аппаратным компонентам, простую электронную аппаратуру (Simple Electronic Hardware, SEH) и сложную электронную аппаратуру (Complex Electronic Hardware, CEH). Важно помнить, что DO-254 был написан

как аналог DO-178B, а не DO-178C, и в то время, когда под “аппаратурой” понималось все “физическое” в системе. Сегодня акцент сместился к более распространенной сложной аппаратной логике, реализованной в ПЛИС, ASIC и других интегральных схемах.

Более поздние публикации, включая консультативный циркуляр FAA (Advisory Circular, AC) 20-152, меморандум CAST-27 и меморандум по сертификации Агентства Европейского союза по авиационной безопасности (European Union Aviation Safety Agency, EASA) CM-SWCEH-001, уточнили, а в некоторых случаях сузили область действия DO-254 и ED-80. В США DO-254 особенно часто применялся с акцентом на заказную сложную электронную аппаратуру, а не на другие виды аппаратуры, перечисленные выше. Почему? Потому что сложная электронная аппаратура использует “логику” на языках описания аппаратуры (Hardware Description Language, HDL), включая VHDL (VHSIC Hardware Description Language, VHDL), а такая логика похожа на программный код. Доказуемые безопасность и качество требуют планирования, стандартов и поэтапных оценок, поскольку последующая верификация менее эффективна, чем процесс, встроенный в разработку.

Кроме того, разработчики авионики в США раньше начали применять к авионике рекомендуемую аэрокосмическую практику SAE ARP4754A, а этот документ требует формальных процессов, требований и испытаний аппаратуры независимо от DO-254. Их коллеги в Европе, Южной Америке и Азии изначально реже применяли ARP4754A, поэтому использовали DO-254, чтобы обеспечить сертификационный охват аппаратуры. Сегодня DO-254 является неотъемлемой частью экосистемы разработки авионики, как показано на следующем рисунке:



Роль DO-254 в экосистеме разработки авионики

Современная электроника использует сложную аппаратуру и ПО, качество которых нельзя полностью оценить только испытаниями конечного изделия. Поэтому важны процессы, с помощью которых разрабатываются соответствующие аппаратура и ПО. Эти процессы должны быть спланированы, оценены и находиться под контролем; DO-254 описывает такие процессы для авиационной аппаратуры. Как показано на предыдущем рисунке, разработка авиационной аппаратуры следует DO-254, но также учитывает окружающие ее мероприятия по оценке безопасности и системной разработке. Инженерия безопасности и системная инженерия участвуют постоянно, чтобы соблюдались требования безопасности и системные требования, учитывалась обратная связь по ним и поддерживалась двунаправленная трассируемость.

DO-254 представляет собой:

- Гибкую основу для разработки бортовой аппаратуры, содержащей функциональность, специфичную для авионики.
- Документ, применимый почти ко всем типам аппаратуры: от датчиков, мультиплексоров, переключателей и объединенных простых интегральных компонентов до полнофункциональных ПЛИС и ASIC.
- Руководство, которое пытается охватить почти бесконечный спектр разновидностей аппаратуры и поэтому неизбежно не дает достаточной конкретики для отдельных проектов.

Как и термин “ПО”, термин “бортовая электронная аппаратура” из названия DO-254 имеет очень широкий охват. В конечном счете аппаратура является частью системы или, точнее, авиационной экосистемы. Поэтому в гражданской авиации применению DO-254 предшествуют оценка безопасности по рекомендуемой аэрокосмической практике ARP4761/A и процесс разработки авиационной системы по ARP4754A. Военная авиация постепенно принимает аналогичный, а в некоторых случаях идентичный, процесс безопасности и системной разработки. Сама аппаратура, как правило, также должна пройти испытания на внешние воздействия по DO-160.

Следовательно, DO-254 является лишь одним звеном в экосистеме сертификации авионики. Нельзя доказуемо показать безопасность и соответствие авиационной аппаратуры без основы безопасности и системной разработки по ARP4761/A и ARP4754A, которая одновременно предшествует DO-254 и сопровождает его.

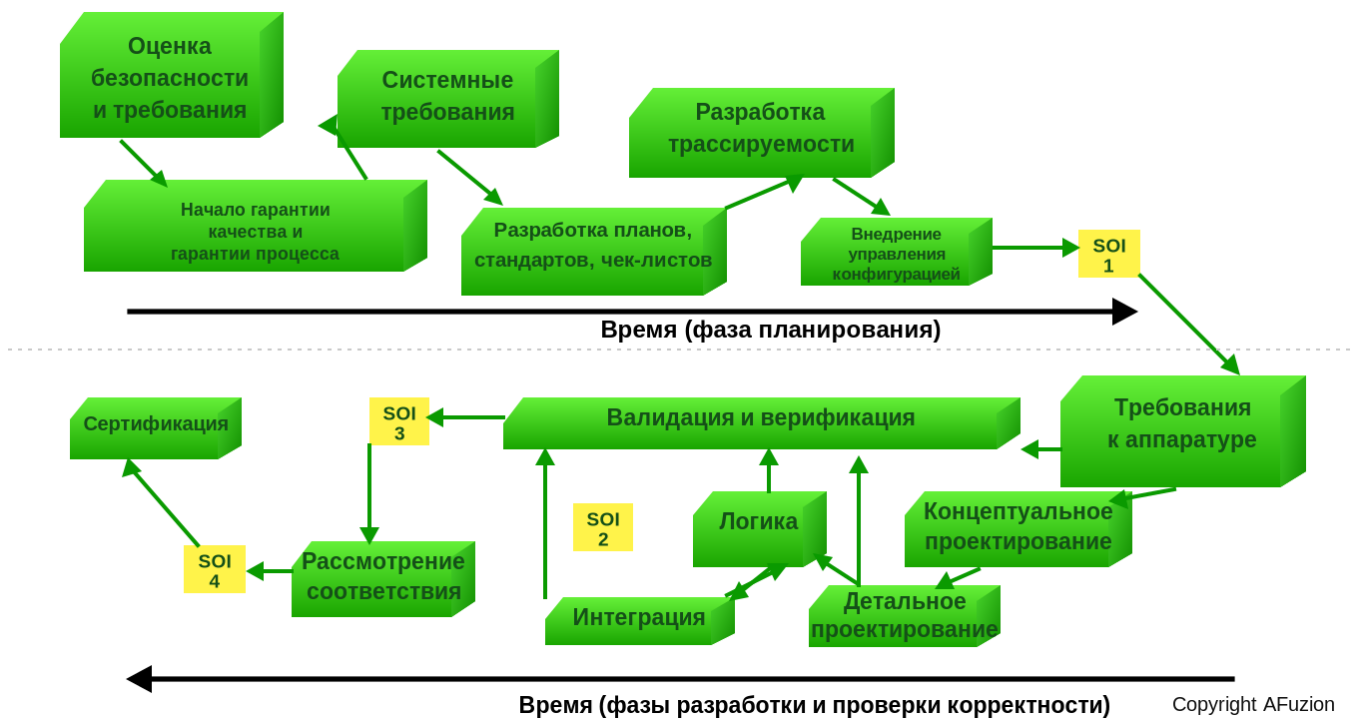
Процесс DO-254 использует детальное проектно-специфичное планирование, за которым следуют постоянные оценки и контуры обратной связи процесса. Это нужно, чтобы определенные процессы разработки аппаратуры, заданные в планах и стандартах, действительно выполнялись. В двух предыдущих главах по DO-178C объяснялись необходимость и содержание пяти планов и трех стандартов. На рисунке и в тексте ниже показано сходство DO-254 с DO-178C:

Три процесса DO-254 и сходство с DO-178C

1. **Процесс планирования.** Планирование выполняется сначала; разработка не начинается до его завершения.
2. **Процесс разработки.** Разработка выполняется после завершения процесса планирования.
3. **Интегральные процессы корректности.** Выполняются непрерывно в течение всего проекта.

Три процесса DO-254 и сходство с DO-178C

Если сильно упростить процесс жизненного цикла DO-254, не показывая необходимые обратные связи, изменения, рассмотрения и т. п., то, по мнению автора, оптимальный инженерный жизненный цикл разработки по DO-254 выглядит следующим образом. Обратите внимание на сходство с ранее показанной “Зеленой змеей” для DO-178C:



Оптимальный маршрут реализации DO-254

Начинаем с безопасности, затем переходим к уровням гарантии разработки.

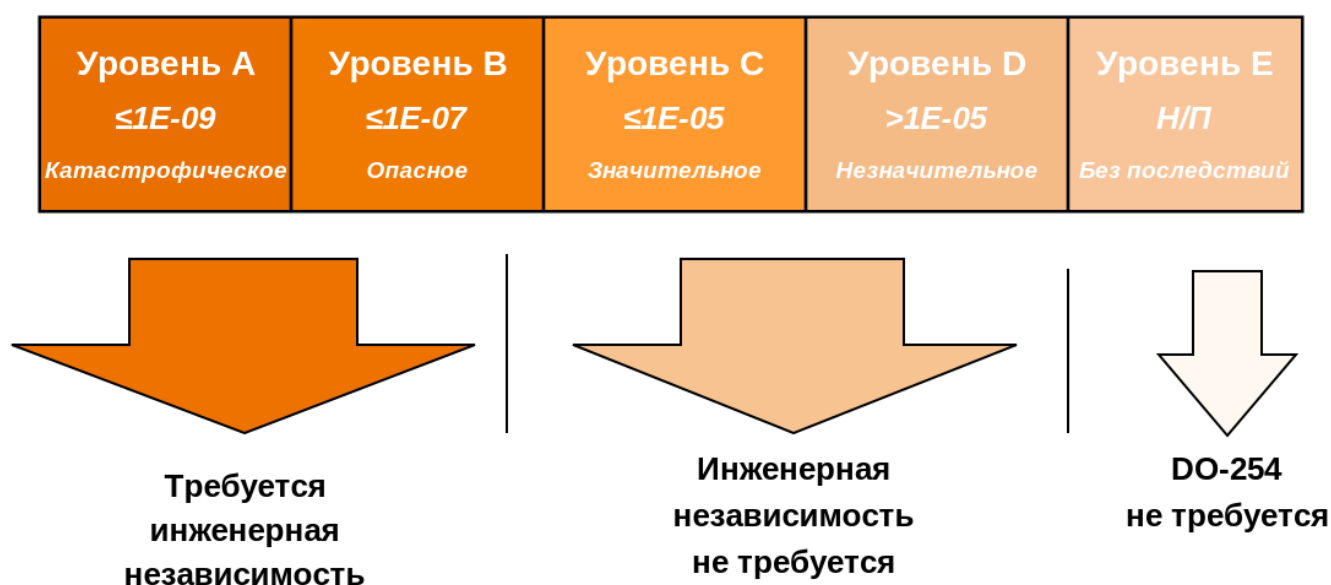
Современные цивилизации, как и некоторые древние, выдвигали теорию “равенства” граждан. В реальной жизни это применимо, но экосистема авионики устроена иначе. Возьмем автомобиль: можно спорить о сравнительном вкладе в безопасность разных систем, например рулевого управления и тормозов, но никто всерьез не станет утверждать, что автомагнитола столь же важна. В авионике различия во вкладе систем в безопасность полетов выражены сильнее и формально регулируются. Для каждой системы обычно выполняется оценка безопасности по рекомендуемой аэрокосмической практике SAE ARP4761A, в ходе которой оценка безопасности и функциональный анализ

аппаратуры задают вопрос: “Какие возможны отказы и каковы их последствия?” Эта оценка безопасности напрямую влияет на системное и аппаратное проектирование, включая резервирование, обнаружение отказов, меры снижения риска и т. п. Фактически она используется при определении архитектуры системы и аппаратуры, а также уровня гарантии разработки (Development Assurance Level, DAL). У каждой уникальной системы есть такой уровень, а аппаратура внутри этой системы, то есть уровень гарантии разработки изделия (Item Development Assurance Level, IDAL), чаще всего имеет критичность, равную уровню системы, то есть уровню гарантии разработки функции (Function Development Assurance Level, FDAL).

DO-254 содержит конкретные цели, зависящие от DAL этой аппаратуры. Для бортовых авиационных систем существуют пять уровней гарантии разработки от А до Е, где уровень А является самым строгим. В DO-178С различия между уровнями для ПО больше, чем в DO-254, и каждому уровню ПО соответствуют явно различные цели: от 26 целей для уровня D до 71 цели для уровня А. Если упростить, уровни DO-254 А и В почти одинаковы: строгие критерии применяются к инженерным процессам, связанным с каждой строкой аппаратной логики. Уровни С и D менее строгие и ориентированы на тщательный подход “черного ящика” к требованиям и испытаниям, без охвата разработки и испытаний аппаратной логики. Уровень Е не требует дополнительной сертификации проектирования аппаратуры по DO-254. Строгость, применяемая к планированию, разработке и корректности аппаратуры, напрямую связана с ее DAL, который часто называют “уровнем критичности”. Эти пять уровней с возрастающей строгостью от уровня Е к наиболее строгому уровню А показаны ниже; обратите внимание на сходство с DO-178С.

Инженерная “независимость” по уровням гарантии разработки (DAL).

Обратите внимание: гарантия качества и гарантия процесса всегда должны быть независимыми



“Независимость” относится к интегральным процессам, например к корректности: верификации, валидации и гарантии процесса. Нужно показать, что в этих интегральных процессах оценку выполняло лицо, отличное от исходного инженера, и что это лицо применяло другой процесс оценки. Например, верификация включает рассмотрения, испытания и анализ.

Как отмечалось ранее, DAL сильно влияет на строгость процесса, применяемого к аппаратуре по DO-254. В следующей таблице обобщены ключевые различия между уровнями DAL для DO-254:

Ключевые различия аспектов уровней гарантии разработки для DO-254

Аспект DO-254	Уровень А	Уровень В	Уровень С	Уровень D
Уровень независимости	Да	Да	Нет	Нет
Необходимость требований к аппаратуре	Да	Да	Да	Да
Необходимость концептуального проекта	Да	Да	Нет	Нет
Необходимость детального проекта	Да	Да	Возможно	Возможно
Покрытие на уровне выводов	Да	Да	Да	Нет
Структурное покрытие операторов	Да	Да	Нет	Нет
Структурное покрытие решений/условий	Да	Да	Нет	Нет
Модифицированное покрытие условий/решений (Modified Condition/Decision Coverage, MC/DC)	Необязательно – рекомендовано FAA	Нет	Нет	Нет
Расширенный метод анализа	Да	Да	Нет	Нет
Управление конфигурацией	Жесткое	Жесткое	Среднее	Низкое
Архивирование аппаратуры и рассмотрения аппаратуры	Да	Да	Низкий уровень	Нет
Соотнесение требований с целевым устройством	Да	Да	Да	Да
Верификация архитектуры и алгоритмов	Да	Да	Да	Нет
Рассмотрения логики/модели	Да	Да	Да	Нет
Критерии перехода гарантии процесса	Да	Да	Да	Нет

DAL аппаратуры обычно, но не всегда, совпадает с DAL системы. Почему? Система содержит аппаратуру, необходимую для выполнения ее функций, и ожидается, что эта аппаратура напрямую связана с вкладом системы в безопасность полетов. Однако этот принцип соблюдается не всегда. Например, автор этой статьи работал над одним проектом DO-254, где система имела DAL B, ПО имело DAL B и DAL D, а ПЛИС имела DAL C. В этом случае ПЛИС лишь преобразовывала пакеты данных, которые проверялись ПО как до нее, так и после нее. Такова интересная особенность правил: к каждому можно найти исключение.

Гарантия процесса похожа на гарантию качества, но имеет более широкий охват, включая аудит поставщиков аппаратуры и аудит перехода к производству. У гарантии процесса есть пять основных видов деятельности, как показано на следующей схеме:

Гарантия процесса в DO-254: пять ключевых ролей

- Вести записи аудитов и метрик.
- Проводить аудит поставщиков аппаратуры.
- Проводить аудит перехода к производству.
- Оценивать соответствие инженерных работ планам, стандартам и контрольным перечням.
- Обеспечивать соответствие планов, стандартов и контрольных перечней проекта DO-254.

Гарантия процесса отличается от гарантии качества ПО, потому что для аппаратуры она должна охватывать аудит поставщиков и обеспечивать, чтобы последующие процессы производства системы были документированы, повторяемы и соответствовали плану. Именно из-за этих двух дополнительных ролей по сравнению с гарантией качества ПО используется другое название: “гарантия процесса”.

Область действия DO-254

DO-254 применяется почти ко всей авиационной аппаратуре, однако более поздние трактовки и практика применения, как отмечалось ранее, сосредоточены на сложной электронной аппаратуре. Почему? Потому что простая аппаратура может быть определена требованиями “черного ящика”, а эти требования, и тем самым вся простая функциональность, могут быть проверены на системном уровне “черного ящика”, например так, как уже требуется по ARP4754A. Для простой аппаратуры значительные дополнительные затраты на детальное планирование, детальное проектирование и низкоуровневую верификацию, предписываемые DO-254, дают мало добавленной ценности. Поэтому они обычно не требуются, если только конкретный сертифицирующий орган не сочтет их необходимыми в отдельном случае, например когда ARP4754A не применялся как самостоятельная основа для системной проверки. Поэтому наиболее распространенное применение DO-254 показано на рисунке ниже; он также поясняет область действия DO-178C:

Линейно-заменяемый блок авионики



Процесс планирования DO-254

Мудрые люди знают: “Что спланировано, то будет сделано”. Более мудрые авиационные органы знают: “То, что тщательно спланировано, можно тщательнее оценить”. Поэтому DO-254 требует детального процесса планирования, состоящего из пяти планов и четырех стандартов:

Процесс планирования DO-254

1. **План сертификации аппаратуры** (Plan for Hardware Aspects of Certification, PHAC).
2. **План процесса гарантии аппаратуры** (Hardware Process Assurance Plan, HPAP).
3. **План управления конфигурацией аппаратуры** (Hardware Configuration Management Plan, HCMP).
4. **План разработки аппаратуры** (Hardware Development Plan, HDP).
5. **План верификации и валидации аппаратуры** (Hardware Verification and Validation Plan, HVVP).

Дополнительно нужны четыре стандарта: стандарт требований к аппаратуре, стандарт проектирования, стандарт архивирования и стандарт верификации и валидации аппаратуры.

План сертификации аппаратуры (PHAC)

План сертификации аппаратуры (Plan for Hardware Aspects of Certification, PHAC) является основополагающим документом планирования для аппаратуры авиационной системы. PHAC дает обзор аппаратуры авиационной системы, критериев безопасности относительно DAL и запланированных сертификационных мероприятий, как показано на следующем рисунке:

План сертификации аппаратуры (Plan for Hardware Aspects of Certification, PHAC)

- Описать систему, аппаратуру, интерфейсы и связь с ПО.
- Сделать документ кратким: как правило, менее 40 страниц; дать обзор мероприятий по аппаратуре.
- Показать соответствие целям DO-254, указать другие планы и стандарты проекта, учесть уровень гарантии разработки (Development Assurance Level, DAL) и вопросы безопасности.
- Описать особые аспекты: квалификацию инструментов, готовые коммерческие изделия (Commercial Off-The-Shelf, COTS), систему на кристалле (System-on-a-Chip, SoC), IP-ядра (Intellectual Property Core, IP Core), историю эксплуатации, проблемные документы и другие специальные темы.
- Включить верхнеуровневый график этапов взаимодействия с сертифицирующим органом (Stage of Involvement, SOI).

План процесса гарантии аппаратуры (HPAP)

План процесса гарантии аппаратуры (Hardware Process Assurance Plan, HPAP) описывает запланированные мероприятия по гарантии процесса в части планирования, аудитов, критериев перехода и рассмотрений соответствия. Обратите внимание: в DO-178C использовался термин “гарантия качества” (Quality Assurance, QA), тогда как в DO-254 эта деятельность называется “гарантией процесса” (Process Assurance, PA). Основные виды деятельности гарантии процесса показаны на следующей схеме:

Основные виды деятельности по гарантии процесса в DO-254

- Ведение записей об аудитах и метриках.
- Аудит поставщиков аппаратуры.
- Аудит перехода к производству.
- Оценка соответствия инженерных работ планам, стандартам и контрольным перечням.
- Обеспечение соответствия планов, стандартов и контрольных перечней проекта DO-254.

Обратите внимание, что два вида деятельности на рисунке выше – аудит поставщиков аппаратуры и аудит перехода к производству – уникальны для роли гарантии процесса в аппаратуре и формально не требуются для роли гарантии качества в DO-178C. Основные темы НРАР обобщены на следующем рисунке:

Основные темы плана процесса гарантии аппаратуры (Hardware Process Assurance Plan, НРАР)

- Описать процессы гарантии процесса (Process Assurance, PA) проекта для DO-254: аудиты, критерии перехода и независимость.
- Описать участие гарантии процесса в планировании, этапах взаимодействия с сертифицирующим органом (Stage of Involvement, SOI) и сертификации.
- Подтвердить независимость гарантии процесса, аудиты и ведение записей.
- Описать оценку гарантии процесса: переходы, поставщиков, производство и рассмотрение соответствия.
- Сделать план универсальным для компании, чтобы его можно было повторно использовать с малыми изменениями.

План управления конфигурацией аппаратуры (НСМР)

План управления конфигурацией аппаратуры (Hardware Configuration Management Plan, НСМР) описывает запланированные мероприятия по управлению конфигурацией, которые обеспечивают установление базовой версии аппаратуры и управление изменениями так, чтобы конфигурацию можно было воспроизвести и повторить. Основные темы этого плана обобщены ниже:

План управления конфигурацией аппаратуры (Hardware Configuration Management Plan, НСМР)

- Определяет все процессы управления конфигурацией аппаратуры.
- Определяет базовые версии, рассмотрения, изменения и выпуски.
- Обеспечивает возможность воспроизвести поставленную аппаратуру в будущем и исключить необоснованные изменения.
- Определяет сообщения о проблемах, управление изменениями, отслеживание и разрешение проблем либо ссылается на соответствующие процедуры.
- Охватывает все артефакты: планы, стандарты, документацию аппаратуры, среду, инструменты, файлы, рассмотрения и аудиты.

План разработки аппаратуры (HDP)

План разработки аппаратуры (Hardware Development Plan, HDP) описывает запланированные мероприятия по требованиям к аппаратуре, концептуальному проектированию, детальному проектированию, реализации и интеграции. Основные темы этого плана обобщены ниже:

План разработки аппаратуры (Hardware Development Plan, HDP)

- Описывает среду разработки, инструменты, процессы и стратегию для требований, проектирования, логики и интеграции.
- Определяет связь и критерии перехода между инженерными этапами.
- Включает все критерии рассмотрений либо ссылается на них: стандарты и контрольные перечни для требований, проектирования, реализации и интеграции.
- Охватывает все аспекты разработки аппаратуры.

План верификации и валидации аппаратуры (HVVP)

План верификации и валидации аппаратуры (Hardware Verification and Validation Plan, HVVP) описывает запланированную валидацию требований к аппаратуре и верификацию аппаратуры: рассмотрения, испытания и анализ применительно к требованиям, проектированию, реализации и интеграции. Помните: валидация подтверждает полноту и корректность требований, а верификация подтверждает, что реализация соответствует этим требованиям. Формально верификация и валидация одинаково важны, однако автор всегда советует считать валидацию более важной: если требования некорректны или неполны, уже неважно, что они верифицированы.

HVVP описывает средства верификации как инженерных решений, заложенных людьми в аппаратуру, так и конечной аппаратуры как результата. Разработка аппаратуры, особенно аппаратной логики в интегральных схемах, требует множества инструментов для проектирования, моделирования, синтеза и изготовления интегральных схем и печатных плат. Все решения, принимаемые людьми при использовании этих инструментов, должны быть документированы и верифицированы. Это нужно не в последнюю очередь для того, чтобы последующие инженеры могли оценить эти решения и в конечном итоге изменить аппаратуру, если потребуются исправления дефектов или новые возможности. HVVP является местом, где документируются эти мероприятия по верификации и валидации. Он обобщен ниже:

План верификации и валидации аппаратуры (Hardware Verification and Validation Plan, HVVP)

- Определяет цели и стратегии верификации требований, проекта, реализации и интеграции.
- Связывает функциональные испытания, испытания робастности, анализ элемента или структуры и места выполнения каждого вида работ.
- Включает контрольные перечни или ссылки на них для всех мероприятий валидации и верификации: рассмотрений, испытаний и анализов.
- Охватывает испытания аппаратуры, тестирование ПО, регрессионный анализ и регрессионное тестирование, а также требования к независимости.
- Описывает критерии перехода для валидации и верификации.

Стандарты

DO-254 благоразумно не предписывает конкретные критерии разработки авиационной аппаратуры: универсальный рецепт был бы непрактичен для множества форм аппаратуры. Вместо этого DO-254 требует, чтобы разработчик аппаратуры определил детальные стандарты для четырех ключевых инженерных этапов, а затем оценивал последующее соответствие этим стандартам. Такая оценка выполняется посредством верификации соответствующих аппаратных артефактов с использованием сохраняемых и управляемых контрольных перечней, которые фиксируют критерии из этих стандартов. Четыре ключевых инженерных этапа разработки аппаратуры по DO-254, требующие стандартов:

Стандарты требований к аппаратуре задают критерии для идентификации, специфицирования и трассировки применимых функциональных и производных требований.

Стандарты проектирования аппаратуры задают критерии для документирования концептуального, а затем детального проектирования аппаратуры, включая все внутренние и внешние интерфейсы.

Стандарты верификации и валидации аппаратуры задают методы, область охвата и критерии валидации, рассмотрений, испытаний и анализа требований к аппаратуре и ее реализации.

Стандарты архивирования аппаратуры задают методы фиксации и хранения аппаратных артефактов с достаточной детализацией, чтобы в будущем можно было воспроизвести все такие артефакты.

Хороший способ понять применение четырех стандартов DO-254 – учитывать, что сам DO-254 охватывает более объективные, то есть измеримые, аспекты жизненного цикла разработки аппаратуры, тогда как некоторые аспекты более субъективны, но не менее важны. К таким субъективным областям относятся требования к аппаратуре,

проектирование аппаратуры, архивирование аппаратуры, а также ее верификация и валидация. Поскольку для каждого проекта авионики нужны собственные процессы и критерии оценки этих субъективных областей, они задаются в четырех проектно-специфичных стандартах. Так эти “субъективные” аспекты становятся “объективными” для конкретного проекта.

Хотя большинство авиационных руководящих документов самодостаточны, DO-254 уникален: он изначально предназначался для охвата всех непрограммных аспектов системы, а сложной аппаратной логики тогда было очень мало. Поэтому со временем к нему добавлялось множество сопутствующих документов, отражающих эволюцию аппаратной среды. Историческая эволюция DO-254 обобщена на следующей схеме:

Историческая эволюция DO-254

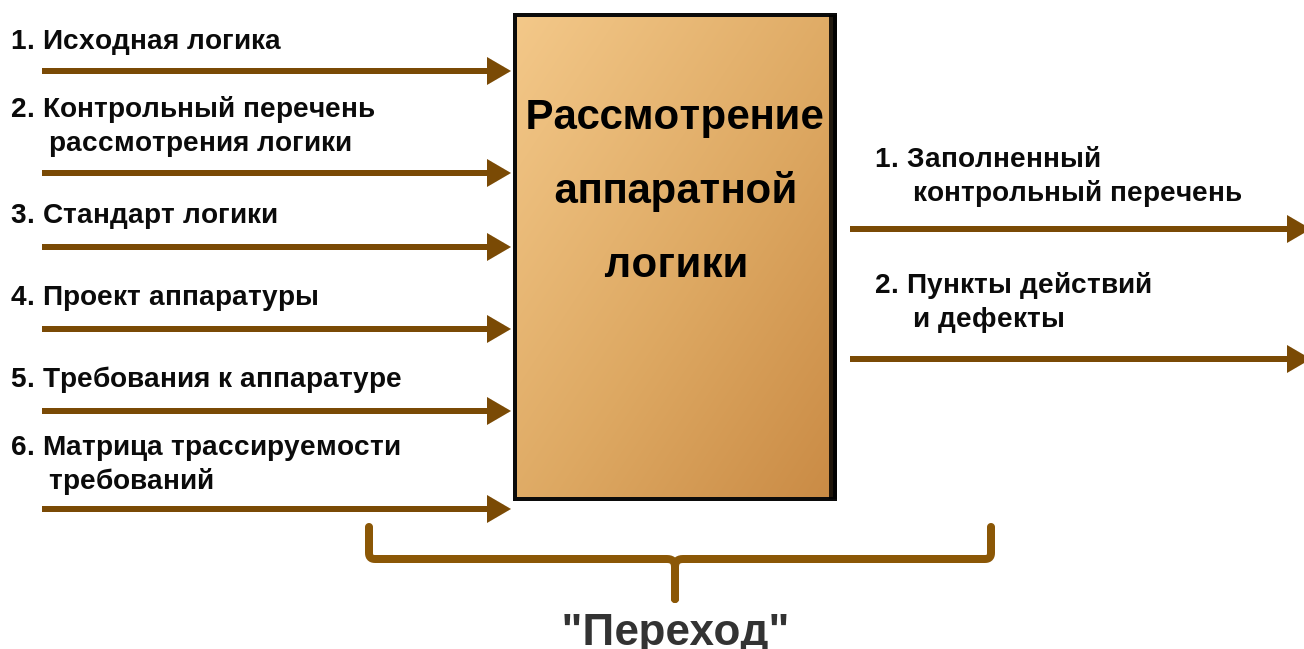
Документ	Год	Основа	Темы
DO-178A	1985	DO-178	Процессы, испытания, компоненты, четыре уровня критичности, рассмотрения, каскадная методология
DO-178B	1992	DO-178A	Интеграция, критерии перехода, различные методы разработки, данные (не документы), инструменты
DO-254	2000 –	DO-178B	Применение процессов типа DO-178B к аппаратуре. Нет четких целей или критериев; более субъективен и вариативен, охватывает всю аппаратуру.
AC 20-152 CAST-27 EASA SWCEN A(M)C 20-152A	2005-2019	DO-254	Уточнение DO-254, требование процессов по типу DO-178C, акцент на логике HDL/VHDL, допущение готовой коммерческой аппаратуры (Commercial Off-The-Shelf Hardware, COTS HW), сужение области действия, чтобы исключить линейно-заменяемые блоки (LRU) и сборки печатных плат, если они охвачены процессами типа ARP4754A, и т. д.

Частый вопрос в проектах по DO-254 звучит так: “Нужны ли аппаратуре и требования высокого уровня (high-level requirements), и требования низкого уровня (low-level requirements), как ПО по DO-178C?” Базовый ответ: “Нет. У аппаратуры только один уровень требований, но два уровня проектирования: сначала концептуальное, затем детальное”. Однако грамотные практики понимают, что формирование требований низкого уровня до реализации ПО уменьшает главную причину дефектов ПО – допущения. Аналогично разработчикам аппаратной логики стоит позаботиться о том, чтобы единственный уровень требований к аппаратуре был достаточно детализирован.

Еще один частый вопрос при применении DO-254 к аппаратной логике, реализованной в интегральных схемах: “Требуется ли DO-254 рассмотрений логики, аналогичных рассмотрениям программного кода в DO-178C?” Краткий ответ: “Да”. Хотя это явно не описано в самом DO-254, поскольку документ появился до массового применения HDL/VHDL в интегральных схемах и ПЛИС, такая информация была добавлена в экосистему DO-254 через документы по вопросам сертификации и консультативные циркуляры. Следующий рисунок показывает необходимые входы и выходы для рассмотрения

аппаратной логики. Обратите внимание на “переходы” от состояния до рассмотрения логики к состоянию после него: они выполняются инженерами и аудируются гарантией процесса аппаратуры.

Необходимые входы и выходы для рассмотрения аппаратной логики:



Общий процесс разработки аппаратуры по DO-254 показан ниже:

Процесс разработки аппаратуры по DO-254



Валидация и верификация аппаратуры (V&V)

Как было описано в предыдущей главе, ПО не валидируется в DO-178C, поскольку валидация требует гарантии “полноты”, которую нельзя получить до тестирования этого ПО на аппаратуре и в составе системы. Термины “валидация” и “верификация” легко ошибочно смешать, но они различны, как показано ниже:

✓ Верификация:

"Соответствует ли реализация требованиям?"

✓ Валидация:

"Корректны ли требования?"

Что же тогда на самом деле означает валидация для аппаратуры? Она весьма похожа на валидацию требований к воздушному судну и системе по ARP4754A. Основные методы валидации аппаратуры задаются в проектном плане верификации и валидации аппаратуры и обобщены ниже:

Основные методы валидации аппаратуры

- Двухнаправленная трассируемость.
- Анализ.
- Моделирование.
- Испытания.
- Сходство.
- Инженерное рассмотрение.
- Для валидации может потребоваться несколько перечисленных методов.
- При валидации должны учитываться как предусмотренные, так и непредусмотренные функции.

Поскольку аппаратный DO-254 мог опереться на опыт DO-178C для ПО, принципы верификации DO-178C в значительной степени применимы и к аппаратуре. Это рассмотрения всех артефактов, связанных с соответствием DO-254, испытания реализации аппаратуры на соответствие требованиям, оценка покрытия логики для DAL A и B, а также дополнительный анализ, когда сочетание рассмотрений и испытаний не дает полностью убедительного результата. Режим верификации DO-254, включающий испытания, рассмотрения и анализ, обобщен на следующей схеме:

Режим верификации DO-254: испытания, рассмотрения и анализ

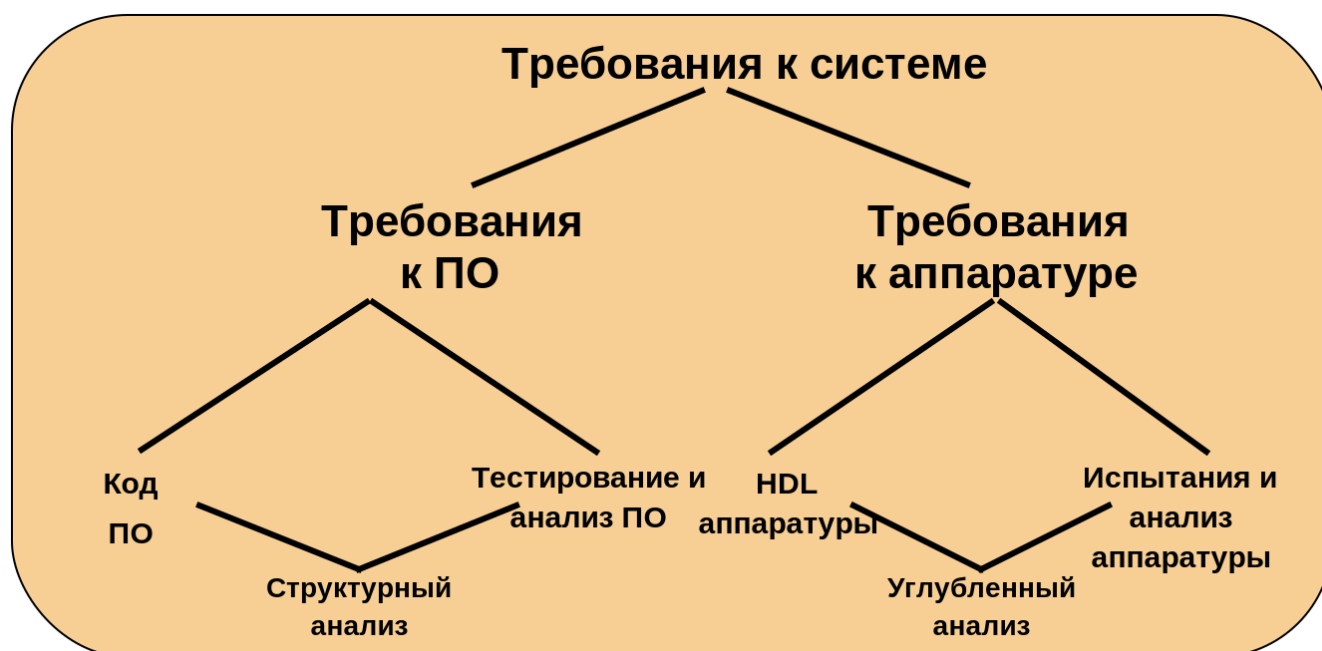
Испытания	Рассмотрения	Анализы
– Испытания на основе требований к аппаратуре – Испытания логики (A/B)	– Требования – Концептуальный и детальный проект – Логика – Результаты испытаний на основе требований – Рассмотрение результатов испытаний низкого уровня (A/B)	– Анализ трассируемости (A/B/C) – Анализ покрытия требований (A/B/C) – Анализ покрытия испытаний низкого уровня (A/B) – Анализ структурного покрытия (A/B) – Общий анализ системы (A/B/C): – Статический временной анализ – Использование устройства – Тепловой режим/питание – Анализ покрытия выводов (A/B/C)

План верификации и валидации аппаратуры должен охватывать следующие дополнительные рекомендации:

- Точно указать, что именно будет **рассматриваться**, когда это будет **рассматриваться**, по каким критериям (ссылка на контрольный перечень) и кем.
- Точно указать, что именно будет **испытываться**, когда это будет **испытываться**, как это будет **испытываться** и кем.
- Указать, какой дополнительный **анализ**, если он нужен, будет выполняться.
- Задавать цели и задачи, а не подробные инструкции “как делать”.
- Давать ссылки на стандарты и контрольные перечни, а не включать их в документ.
- Охватывать валидацию производных требований.
- Описывать различные формы верификации и среду для каждой из них.
- Делать документ общекорпоративным: пригодным для повторного использования в других проектах с незначительными изменениями.

Трассируемость аппаратуры в сравнении с ПО

Еще один типичный вопрос касается трассируемости: “Нужна ли в DO-254 двунаправленная трассируемость и похожа ли она на трассируемость ПО?” И снова краткий ответ: “Да”. Следующая схема обобщает трассируемость для аппаратуры в сравнении с ПО:



Трассируемость аппаратуры и ПО

Как показано на схеме выше, трассируемость всегда является двунаправленной и охватывает системные требования, требования к аппаратуре и ПО, проверки, а также реализацию: программный код и аппаратную логику. При этом концептуальное проектирование аппаратуры и архитектуру ПО не требуется включать в трассируемость как отдельные явные элементы. Тем не менее трассировка от требований к логике проходит через проектирование; важно, чтобы проектные решения не скрывали и не разрывали эту трассируемость.

Готовая коммерческая аппаратура (COTS) и IP-ядра

В то время как использование готового коммерческого ПО (Commercial Off-The-Shelf, COTS) в авиации обычно требует полного применения DO-178C, готовая коммерческая аппаратура встречается в авиации чаще и легче используется в сертифицируемых системах. Вместо перепроектирования аппаратуры только ради соответствия DO-254 преобладающая стратегия сертифицирующих органов такова: “Не вносить новые ошибки через новые проекты аппаратуры, а сохранять проверенные электронные компоненты, демонстрируя при этом надлежащее инженерное управление этой аппаратурой”. Такое инженерное управление демонстрируется с помощью плана процесса управления электронными компонентами (Electronic Component Management Process Plan, ECMP). Ниже обобщены ECMP и его применение:

Сводка и применение плана процесса управления электронными компонентами (Electronic Component Management Process Plan, ECMP) для готовой коммерческой аппаратуры (Commercial Off-The-Shelf Hardware, COTS HW)

- Обеспечить наличие процесса управления электронными компонентами.
- Процесс управления электронными компонентами охватывает выбор, квалификацию и управление конфигурацией готовых коммерческих устройств, включая неиспользуемую функциональность.
- Процесс управления электронными компонентами охватывает доступ к данным компонентов: руководству пользователя, техническому описанию, перечню ошибок и руководству по установке.
- Процесс управления электронными компонентами охватывает доступ к информации об изменениях, внесенных изготовителем компонента.
- Процесс управления электронными компонентами учитывает зрелость готового коммерческого устройства и меры снижения рисков.
- Не использовать устройство вне пределов технического описания. Иначе необходимо отдельно оценить надежность.

Сводка и применение ECMP для готовой коммерческой аппаратуры (COTS)

Еще один часто задаваемый вопрос по DO-254 касается IP-ядер (Intellectual Property Cores, IP Cores; здесь IP не означает Internet Protocol): “Есть ли в DO-254 специальные правила для IP-ядер, или можно просто использовать любое такое ядро?” На этот вопрос нет простого ответа “да” или “нет”, потому что IP-ядра практически не существовали в авиации в то время, когда DO-254 составлялся в конце 1990-х годов. Однако они становятся все более распространенными и дают множество преимуществ, включая повторное использование, качество и сокращение времени вывода на рынок, подобно библиотекам ПО и операционным системам реального времени, OSCPВ (Real-Time Operating System, RTOS). IP-ядра по сути делятся на жесткие, мягкие и полужесткие, как обобщено на следующей схеме:

Аппаратные IP-ядра (Intellectual Property Core, IP Core): жесткие, мягкие и полужесткие

- **Жесткое IP-ядро.** Оптимизировано под конкретную реализацию в интегральной схеме; менее переносимо.
- **Мягкое IP-ядро.** Представлено исходной логикой на языке описания аппаратуры (Hardware Description Language, HDL); более переносимо.
- **Полужесткое IP-ядро.** Промежуточный вариант между мягким и жестким IP-ядром; часто поставляется в формате списка соединений.

Аппаратные IP-ядра: жесткие, мягкие и полужесткие

Хотя готовые коммерческие IP-ядра, как правило, уже прошли отладку в эксплуатации, а изготовители заинтересованы в качестве продукции, они все равно должны подвергаться тщательным испытаниям на основе требований, включая проверку робастности. Для DAL A и B требуется дополнительная поддержка со стороны изготовителя, чтобы доказуемо показать применение внутренних процессов разработки и управления конфигурацией.

Эволюция разработки ПО включала разработку на основе моделей и объектно-ориентированные технологии, которым посвящены последующие главы книги. При этом базовое применение языка высокого уровня, компилятора и компоновщика в разработке ПО достаточно устоялось и мало изменилось. С аппаратурой все иначе: ее эволюция в авиации была крайне значительной. В предыдущих главах этой книги объяснялись сравнительно небольшие изменения в сертификации ПО от DO-178A до DO-178C. Очевидно, DO-254 очень выиграл бы от более частых обновлений или хотя бы от одного обновления; этого так и не произошло. Вместо этого “применение” сертификации аппаратуры на основе DO-254 изменялось через более частые вспомогательные сертификационные меморандумы, такие как AC 20-152, CAST-27, EASA CM-SWCEH-001 и AMC 20-152A. Последний был подготовлен в 2019 году и одобрен в 2020 году. Соответствующие ключевые разъяснения из этих последующих вспомогательных документов и AMC 20-152A обобщены на следующей схеме:

Развивающиеся разъяснения DO-254, включая приемлемый метод определения соответствия AMC 20-152A (Acceptable Means of Compliance, AMC)

- **Уровень гарантии разработки D (Development Assurance Level, DAL D) и узлы печатных плат.** DO-254 не требуется.
- **Готовая коммерческая аппаратура (Commercial Off-The-Shelf, COTS).** Допустим упрощенный подход как к "черному ящику", но нужны полное определение функций, тщательные испытания функционального определения и строгое управление конфигурацией COTS.
- **Готовые коммерческие IP-ядра.** Для них применяются более строгие и детальные критерии: выбор и оценка поставщика, планирование, детальные и производные требования, интеграция, валидация и верификация.
- **Дополнительные обязательные темы.** План управления устареванием, снижение риска одиночных сбоев (Single Event Upset, SEU), производные требования и обратная связь.

Авиационные требования к системам, ПО и аппаратуре

Автор: Вэнс Хилдерман

Рецензент:

Г-н Marc GATTI, научный директор Thales AVS France SAS

Хорошие требования – основа хороших изделий, а единственный путь к “великому” изделию проходит через великие требования. В аэрокосмической отрасли, или в авионавтике, требования важны для всех аспектов изделия: аппаратуры, ПО, систем, безопасности полетов, а теперь еще и информационной безопасности. Но как “предписать” хорошие требования? Что должен содержать аэрокосмический или авионический стандарт требований? Как выглядят слабые, приемлемые, хорошие и, наконец, великие требования? Отличаются ли требования к бортовым и наземным системам, к ПО и аппаратуре? Жизнеспособность авиационной отрасли требует ответов на эти вопросы. Итак, вот они.

Для начала рассмотрим небольшую авиационную “викторину” по требованиям. Знаете ли вы ответы? Если вы разрабатываете авиационные системы, знать их необходимо. Ответы на эти и многие другие вопросы рассматриваются в этой и следующих главах.

- 1. Верно/неверно: в большинстве авиационных проектов большая часть требований относится к безопасности.**
- 2. Верно/неверно: разработка авиационных требований должна следовать каскадной модели.**
- 3. Верно/неверно: лучший способ оценивать авиационные требования – выполнять тестирование.**
- 4. Верно/неверно: рекомендуемые аэрокосмические практики серии ARP47XX (Aerospace Recommended Practice, ARP) и документы DO-XXX дают четкие указания по разработке и оценке требований.**
- 5. Верно/неверно: большинство дефектов авионики вызвано ошибками в коде и производственными дефектами.**
- 6. Верно/неверно: при разработке на основе моделей требования должны быть текстовыми и находиться вне модели.**

Если спросить сертифицирующие органы: “Что важнее: требования к бортовым авионическим системам, аппаратуре, ПО или наземным системам?”, то самый верный и, скорее всего, самый ожидаемый ответ будет: “Да”. Любой другой ответ мог бы ошибочно навести спрашивающего на мысль, что один вид требований нужно ставить выше другого. Как отмечают многие эксперты по разработке критичных для безопасности систем, а также книги и статьи автора, требования являются основой авиационной разработки. Эксперты по критичным для безопасности системам во всем мире постоянно указывают, что главная причина дефектов, связанных с безопасностью, прослеживается к слабым требованиям. Слабые требования означают опору на допущения, причем разные инженеры делают разные допущения. Если допущения различаются, значит по крайней мере одно из них, а возможно и все они, **неверны**.

Однако все авиационные руководящие документы, на которые опираются авиационные сертифицирующие органы, довольно расплывчаты в части методик оптимальной разработки требований. Многие ошибочно считают это слабым местом документов, но на самом деле причин три:

- 1. Авиационные системы сильно различаются по функциональности, критичности и выбранным методологиям разработки; поэтому единого руководства по написанию требований было бы недостаточно.*
- 2. Вместо этого используется обязательный подробный “Стандарт требований”, который готовится для каждого авиационного проекта и описывает проектный подход к разработке требований, а также критерии оценки таких требований.*
- 3. Талантливые первоначальные разработчики меньше нуждаются в подробных требованиях, чем простые смертные. Но подавляющее большинство авиационных систем не являются “первоначальными”: это очередные модернизации и варианты прежних систем. Поэтому изменения вносит не создатель, а менее осведомленные инженеры. Верификаторы, аудиторы и сертифицирующие специалисты этих систем тоже никогда не обладают тем же знанием изделия, что и первоначальный разработчик. Все эти люди в значительной степени полагаются на требования, чтобы понять и оценить авиационное изделие.*

Несколько слов о верификации и валидации (Verification and Validation, V&V)

Представьте разговор между слишком типичным авиационным инженером и человеком не из инженерной среды:

Неинженер	
Неинженер:	“Что вы сегодня делали?”
Инженер:	“Я занимался V&V.”
Неинженер:	“Что такое V&V?”
Инженер:	“Verification and Validation. То есть верификация и валидация. Или, может быть, валидация и верификация.”
Неинженер:	“Вы занимались валидацией или верификацией?”
Инженер:	“Да, я занимался V&V.”
Неинженер:	“Но все-таки валидацией или верификацией?”
Инженер:	“Ну, разницы никто не знает, поэтому мы просто называем это V&V!”
Неинженер:	“Инженеры странные...”
Инженер:	“Да.”

На самом деле верификация и валидация просты

Верификация: соответствует ли реализация требованиям? Валидация: являются ли требования “правильными”?

Теперь спросите сертифицирующие органы, что важнее: верификация или валидация. Политически корректный ответ снова будет “Да”, потому что верификация не должна быть менее важной, чем валидация, и наоборот. Но если спросить действительно опытного авиационного инженера, единственный хороший ответ будет таким: “Валидация, потому что без хороших требований уже неважно, насколько хорошо вы их верифицировали!”. Это совершенно верно. Но что такое “хорошие требования”? Валидация требований подтверждает, что они корректны и полны. Для авиационного ПО валидация выполняется через аппаратуру на системном уровне, поэтому требования к ПО технически валидируются путем рассмотрения. Здесь существенно помогает моделирование по DO-331, *Дополнению по разработке и верификации на основе моделей к DO-178C и DO-278A* (Model-Based Development and Verification Supplement to DO-178C and DO-278A): оно обеспечивает видимость требований и валидацию на нескольких уровнях, от заказчика до системы, аппаратуры и ПО. Для требований к аппаратуре и системе валидация включает рассмотрение, имитационное моделирование, моделирование, двунаправленную трассируемость, анализ и любую другую комбинацию методов, необходимых для подтверждения полноты и корректности.

Поскольку сложность авиационных систем постоянно растет, одного уровня требований недостаточно. Возможно, ранней авиации хватало одного уровня требований, но рост сложности и увеличение инженерных команд повышают вероятность ошибочных

допущений. Напомним вывод из предыдущих глав: причина номер один дефектов авиационных систем, ПО и аппаратуры – это “допущения”. Лучший способ уменьшить количество допущений – улучшить требования. Поэтому авиационные системы имеют несколько уровней требований, включая:

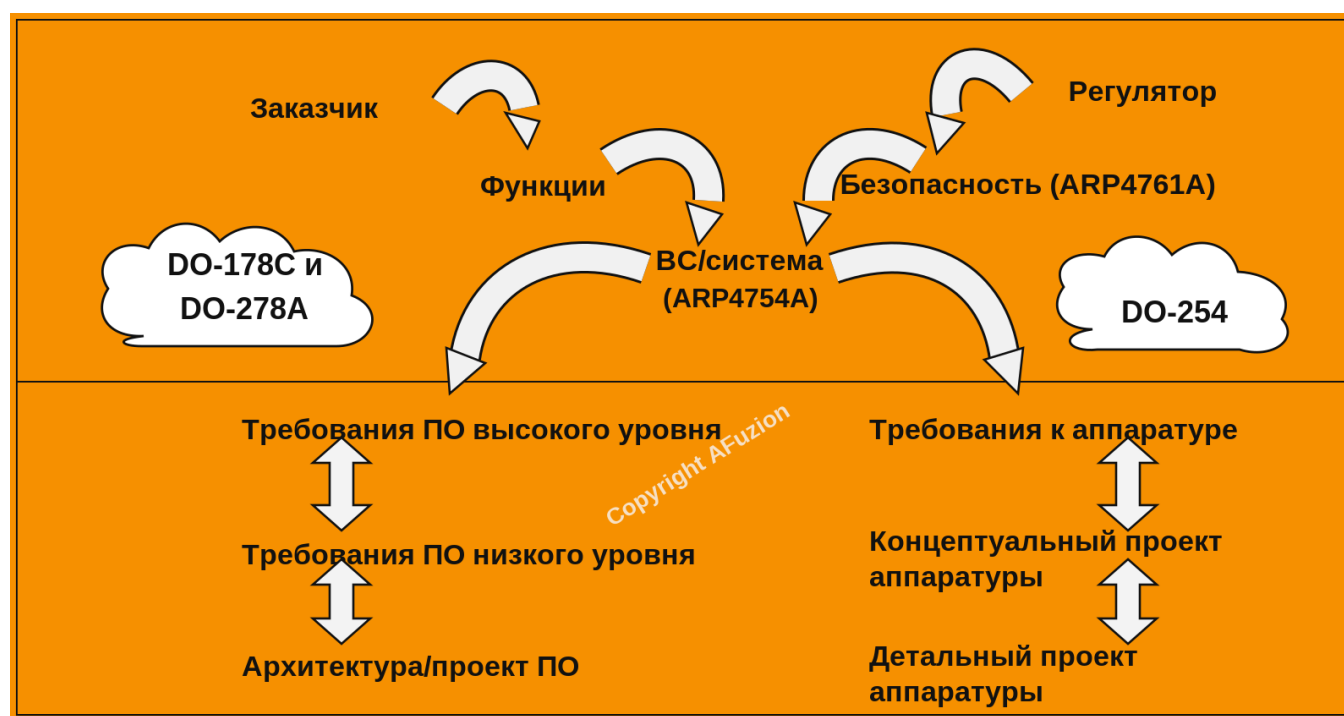
Типичные уровни авиационных требований

1. Верхнеуровневые требования к воздушному судну по рекомендуемой аэрокосмической практике ARP4754A (Aerospace Recommended Practice, ARP) либо требования к крупномасштабной наземной системе, включая требования, заданные DO-278A.
2. Требования безопасности по ARP4761A; также могут включать требования к информационной безопасности по DO-326A.
3. Требования к воздушному судну и системе по ARP4754A.
4. Требования к аппаратуре по DO-254.
5. Требования к ПО по DO-178C и DO-278A, а также требования к аэронавигационным данным по DO-200B.

Каждая из приведенных выше областей требований обычно соответствует все более детальному уровню в показанном порядке. На всех уровнях могут декомпозироваться или выводиться дополнительные требования по безопасности, которые уточняют необходимые свойства системы, аппаратуры и ПО. Если система особенно сложна, любую из этих областей можно дополнительно разделить на два или более уровня требований. В результате появляется несколько уровней требований, которые повышают качество: связи между требованиями становятся понятнее, сами требования проще валидировать, а затем верифицировать. Разработка авиационных требований означает последовательную декомпозицию с рассмотрением требований на каждом этапе уточнения. Для высоких уровней гарантии разработки, связанных с опасными или катастрофическими последствиями отказов, необходимо доказать независимость верификации и валидации требований. Например, их должен выполнять другой человек или другая команда по процессу, независимому от разработчика требований.

Для ПО и аппаратной логики все чаще характерна ситуация, когда сложность или размер ПО настолько велики, что ни один человек не понимает больше небольшой части этой логики. Поэтому для ПО бортовой авионики по DO-178C и наземных или спутниковых систем связи, навигации, наблюдения и организации воздушного движения (Communication, Navigation, Surveillance / Air Traffic Management, CNS/ATM) по DO-278A по умолчанию требуются два уровня требований: требования высокого уровня (High-Level Requirements, HLR) и требования низкого уровня (Low-Level Requirements, LLR). Хорошие разработчики аппаратуры, работающие по DO-254, часто применяют два или более уровня требований как лучшую практику, хотя формально DO-254 этого не требует.

Системные требования должны охватывать всю функциональность системы на уровне “черного ящика”, включая требования по безопасности. Затем системные требования декомпозируются и распределяются на аппаратуру, ПО или на то и другое. Требования высокого уровня к ПО определяют, что ПО делает на уровне интерфейса аппаратуры и ПО; они не должны содержать деталей уровня программных функций или модулей, поскольку эти детали задаются требованиями низкого уровня. Хорошей практикой считается выделять интерфейсы аппаратуры и ПО (Hardware/Software Interfaces, HIS) в отдельный документ. Причина двух уровней требований фундаментальна: простая функциональность ПО может быть полностью описана одним уровнем требований, но авионическое ПО сложно и становится еще сложнее. Поэтому выявление требований, реализация и верификация надежнее выполняются тогда, когда спецификация проходит две или более стадии уточнения. На рисунке ниже показана экосистема авиационных требований. Обратите внимание: производные требования тоже должны быть включены, поскольку они необходимы для учета аспектов безопасности, например разнородности и мониторинга технического состояния, а также для закрытия пробелов. Иначе не вся функциональность будет охвачена требованиями, а значит не сможет быть полностью трассирована и верифицирована. Такие производные требования не обязательно трассируются к родительскому требованию, поэтому с точки зрения безопасности им требуется дополнительное независимое рассмотрение.



Экосистема авиационных требований

В более ранних версиях авиационных руководящих документов, например DO-278 и DO-178B, формальные цели, связанные с требованиями высокого и низкого уровней, были особенно расплывчаты. Создавалось впечатление, что формального требования “вернуться и улучшить требования” нет. DO-178C и DO-278A это исправили: они предписали, чтобы требования низкого уровня были детализированы вплоть до ветвей

логики, и фактически заставили улучшать требования, обычно через уточнение требований низкого уровня, когда анализ структурного покрытия выявляет непокрытые структуры кода из-за слабых требований низкого уровня. Поэтому автор и его сотрудники часто видели довольно слабые требования низкого уровня в унаследованных системах, особенно в наземных системах CNS/ATM.

Терминология требований

У авиационных требований своя таксономия, несколько отличающаяся от неавиационных отраслей. Тонкости ключевых терминов и их применения подробно рассмотрены в следующих абзацах, а на рисунках ниже дано краткое резюме:

Терминология авиационных требований, часть 1

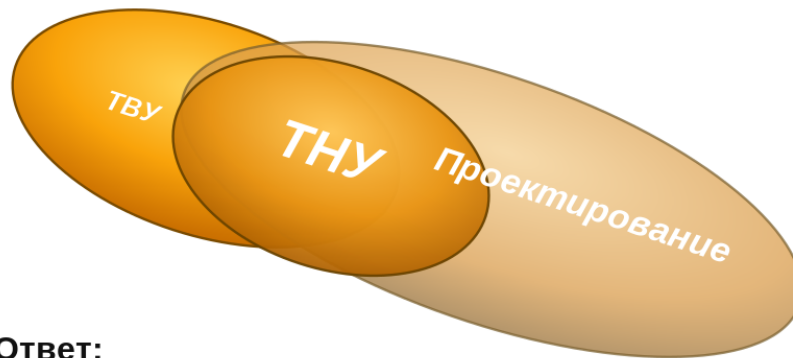
- **Требование высокого уровня.** Требование, трассируемое к системному требованию, которое указывает, что должно делать ПО, а не как оно должно это делать.
- **Производное требование.** Требование, возникшее из проектного решения; оно может не трассироваться назад напрямую, но должно трассироваться вперед.
- **Декомпозированное требование.** Системное требование или требование высокого уровня, удовлетворяемое комбинацией двух или более требований более низкого уровня.

Терминология авиационных требований, часть 2

- **Описание проекта.** Сочетание требований низкого уровня и архитектуры; может включать словарь данных.
- **Требования низкого уровня.** Требования, которые декомпозируют требования высокого уровня, чтобы показать, как они выполняются.
- **Архитектура.** Схема, показывающая, как компоненты системы связаны между собой; обычно представляется с точки зрения управления и потоков данных.

Схема из предыдущей главы снова приведена ниже, чтобы подчеркнуть связь между стадиями требований к ПО и проектированием:

Требования низкого уровня: что это?



* **Ответ:**

**Перекрытие требований высокого уровня
и проектирования = требования низкого уровня**

Авиационные руководящие документы не дают рецепта составления требований и не проводят четкую границу между требованиями высокого и низкого уровней, а также между требованиями низкого уровня и проектом. Вместо этого каждый проект должен предоставить проверяемый “рецепт” в виде формального стандарта требований, по которому требования будут оцениваться. Лучшей практикой считается сначала полностью определить требования (высокого уровня и, при необходимости, низкого уровня), а уже потом переходить к проектированию. В любом случае проект нужно рассматривать с учетом всех связанных требований, как высокого, так и низкого уровня. Помните: все требования подлежат тестированию, а проект не тестируется, а документируется и рассматривается. Поскольку все требования тестируются независимо от их отнесения к высокому или низкому уровню, фактическая граница между “высоким” и “низким” менее важна. Если у вас есть идеальная декомпозиция требований высокого уровня в требования низкого уровня и вы полностью тестируете требования низкого уровня, нужно ли все равно тестировать требования высокого уровня? Ответ: да. И требования высокого уровня, и требования низкого уровня являются входами процесса рассмотрения тестов, поэтому при рассмотрении тестов нужно учитывать оба уровня. В реальной жизни “идеальных” требований не бывает, а тестовые примеры следует разрабатывать до проектирования и реализации ПО, чтобы их рассмотрение помогало улучшать сами требования.

Требования высокого уровня (HLR)

Разработка требований высокого уровня следует за определением системных требований и предшествует разработке требований низкого уровня. Требования высокого уровня должны быть формально рассмотрены, см. ниже, до разработки связанных с ними требований низкого уровня. Следующий рисунок показывает ключевые причины, по которым требования высокого уровня необходимы для любой авиационной системы, потенциально влияющей на безопасность полетов и информационную безопасность и потому требующей сертификации ПО и аппаратуры:

Ключевые причины разрабатывать требования высокого уровня

- Определять функциональность до начала проектирования.
- Обеспечивать трассируемость к системным требованиям.
- Выводить дополнительную функциональность ПО.
- Трассировать тестирование к требованиям, а не к коду.
- Уменьшать число изменений в интерфейсах.
- Снижать число допущений.

Требования высокого уровня – это “мост” между системными требованиями и требованиями низкого уровня. Хорошие требования высокого уровня должны охватывать и прояснять следующие аспекты ПО:

Атрибуты, охватываемые требованиями высокого уровня

- Функциональность.
- Внешние интерфейсы аппаратуры и ПО.
- Характеристики.
- Атрибуты качества.
- Проектные ограничения.
- Безопасность полетов и информационная безопасность.

Требования высокого уровня, а также рассматриваемые ниже требования низкого уровня, которые устанавливаются путем анализа или декомпозиции системных требований либо архитектуры системы, называют производными из-за их прямой связи с системой. Требования высокого уровня, происходящие из требований, связанных с безопасностью, обычно тоже называют производными. Но само различие “производное/непроизводное” здесь менее важно, потому что такое требование высокого уровня наследует атрибут безопасности от своего источника и должно быть возвращено в процесс обеспечения безопасности. В этом и состоит реальная причина существования “производных” требований и требований, связанных с безопасностью, в ARP4754A и DO-178C, но не в прежнем DO-178B: все такие требования должны независимо рассматриваться специалистами по безопасности как часть формального процесса обратной связи по безопасности. Этот процесс также должен аудироваться службой гарантии качества ПО (Software Quality Assurance, SQA) и службой гарантии системного процесса (Systems Process Assurance). Требования высокого уровня, полученные при анализе материалов оценки безопасности, а не при декомпозиции уже заданных требований, связанных с безопасностью, всегда являются производными: у них нет родительского требования. В системе управления требованиями для них также должен быть установлен атрибут безопасности.

Требования низкого уровня (LLR)

Требования низкого уровня следуют за определением требований высокого уровня и предшествуют проектированию либо сопровождают его. Требования низкого уровня должны быть формально рассмотрены, см. ниже, до разработки соответствующей логики. Следующий рисунок кратко показывает, почему требования низкого уровня необходимы для любого авиационного ПО, потенциально влияющего на безопасность полетов и/или информационную безопасность и потому требующего сертификации ПО и аппаратуры:

Ключевые причины разрабатывать требования низкого уровня

- Завершать определение функциональности до выхода из проектирования.
- Обеспечивать трассируемость к требованиям высокого уровня.
- Выводить дополнительную функциональность и свойства робастности.
- Повышать вероятность корректного кодирования с первого раза.
- Трассировать тестирование требований низкого уровня к требованиям, а не к коду.
- Уменьшать число изменений при разработке и в производстве.
- Сокращать затраты и сроки.

Требования низкого уровня – это “мост” между требованиями высокого уровня и проектированием/реализацией. Хорошие требования низкого уровня должны прояснять следующие аспекты ПО:

Атрибуты, охватываемые требованиями низкого уровня

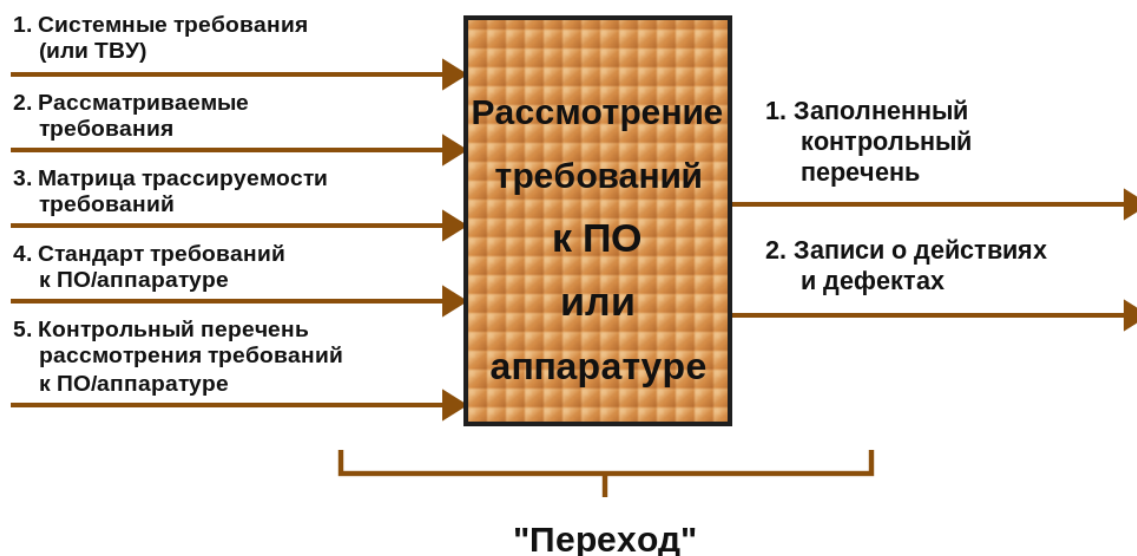
- Уточнение требований высокого уровня.
- Внутренние интерфейсы ПО.
- Параметры функций и методов.
- Свойства робастности.
- Большинство логических решений.
- Все случаи времени выполнения.

Авиация: исправлять требования или исправлять изделие?

Ключевая особенность авиационных требований – ранняя валидация. Валидация оценивает корректность и полноту требований. Поскольку валидация требует учета аппаратуры, требования нужно валидировать с учетом системных и аппаратных аспектов. Все требования должны быть верифицированы: верификация состоит из формальных рассмотрений, а позднее и тестирования. Если тестирование не дает окончательного вывода, может потребоваться дополнительный анализ. Очевидно, что требования лучше улучшать до реализации: предотвращение дефектов дешевле, чем их позднее

исправление. Верификация, в свою очередь, задает вопрос: “Соответствует ли реализация требованию или требованиям?” Все требования должны формально рассматриваться. Здесь “формально” означает доказуемое применение критериев рассмотрения вместе с ведением записей и исправлением дефектов. Например, рассмотрение требований к авиационному ПО или аппаратуре имеет такие входы и выходы:

Входы и выходы рассмотрения требований



Как показано выше, у формального рассмотрения требований есть пять входов. Все пять должны находиться под управлением конфигурацией, и должно быть доказано, что они использовались при фактическом рассмотрении. Эти пять входов, включая производные требования, если они применимы, образуют входные критерии формального рассмотрения требований. Заполненный контрольный перечень рассмотрения требований и записи о действиях и дефектах образуют выходные критерии. Движение от входа мероприятия к выходу образует “переход”. Верификатор требований выполняет переход, а служба гарантии качества аудирует его. Для рассмотрений количество рецензентов незначительно. Фактически опыт автора показывает, что лучшие рассмотрения получаются при меньшем числе более сильных рецензентов; в сложных системах размер команды **как правило обратно пропорционален** итоговому качеству и уж точно производительности. Ключ к рассмотрению требований – применение соответствующего стандарта требований и контрольного перечня рассмотрения. Типичные качественные стандарты требований для критичных для безопасности систем подробны и занимают более 20 страниц; качественные контрольные перечни рассмотрения требований также подробны и обычно занимают 6-8 страниц или больше. Это резко контрастирует с изделиями, не критичными для безопасности: там стандарты требований и контрольные перечни часто отсутствуют, а если присутствуют, то остаются очень поверхностными.

В ранних авиационных руководящих документах необходимость и качество требований признавались, но обеспечивались слабее. Современные документы требуют постоянного уточнения требований так, чтобы и дефекты, и недостатки покрытия при тестировании логики в первую очередь устранялись повышением ясности требований. DO-178C и DO-278A пошли еще дальше: они требуют, чтобы тесты, подтверждающие

покрытие кода ПО, то есть структурное покрытие, основывались на требованиях. Это означает, что требования низкого уровня должны быть очень подробными и давать достаточно информации, чтобы использовать их при рассмотрении кода для логических решений.

Производные требования: английский против “авилиша”

Много лет назад автор устроил праздничную вечеринку более чем для ста своих сотрудников – авиационных инженеров из разных стран. Его супруга заметила, что инженеры говорили на своеобразной смеси английского, сильно пропитанной авиационной терминологией, значительную часть которой она не понимала. Они решили, что даже носители английского на самом деле говорят не по-английски, а скорее на “авилише” (Avilish), то есть на авиационном английском (Aviation English). Это стало внутри компании жаргонным названием авиационного английского. В “авилише” обычные английские слова получают совсем иной смысл. Вероятно, самое значимое превращение авиационного английского связано с термином “производное требование” (derived requirement). В нормальном английском, и даже в американском английском, derived означает “происходящий от чего-то”. Например, “ДНК человека происходит от его родителей” или “оптимальный маршрут можно вывести, изучив карту”. На этом обычный английский заканчивается...

В авиации “производное требование” определяется как “требование, выведенное из процесса оценки безопасности или инженерного решения, чтобы предписать определенную функциональность или возможность, которая может не быть связана ни с какими родительскими требованиями, и закрыть функциональные пробелы, которые иначе нельзя декомпонировать из родительских требований”. Критичные для безопасности системы почти всегда имеют аспекты, связанные с безопасностью, которые прямо или косвенно повышают безопасность: резервирование, встроенный контроль, мониторинг технического состояния и т. д. Все это является функциональностью, связанной с безопасностью, но для нее может не быть родительских требований, прямо предписывающих такие конкретные аспекты. Простая аналогия – офисное здание. Скорее всего, в вашем офисе есть дымовые извещатели, спринклерная система и запасная дверь. Все они действительно выполняют свои функции, но они связаны с безопасностью и функционально не нужны для обычного использования офиса. Будем надеяться, дымовые извещатели, спринклеры и запасная дверь никогда не понадобятся. Но, как и требование иметь ремень безопасности и подушку безопасности в автомобиле, а это тоже требования по безопасности, они выводятся из оценки безопасности или нужны для выполнения заданных требований по безопасности. Поэтому в авиации их называют “производными”.

Следовательно, в авиации такие аспекты задаются производными требованиями согласно приведенному выше определению. Примеры аспектов авиационных производных требований включают:

- требования по безопасности, например мониторинг технического состояния для сброса, деактивации или переключения;
- частоты выборки системы;
- резервирование;
- регистрацию и выдачу сообщений об отказах;
- архитектурные решения: триплекс вместо двухканальной схемы с мониторингом;
- частоту стробирования сторожевого таймера;
- разнородность;
- определение входов и выходов аппаратуры/ПО.

Все перечисленное можно было бы назвать “функциональными требованиями”, просто добавив для них родительские требования. Но требования, связанные с безопасностью, должны быть явно возвращены в процесс обеспечения безопасности для официального и документированного рассмотрения. Если вместо этого обозначить их как функциональные, можно обойти обязательное рассмотрение по безопасности. Так делать не нужно. Называйте жеребенка жеребенком, а не лошадью, и обеспечивайте ему надлежащее внимание и рассмотрение: “производное требование по безопасности”.

Требования = “что”, проект = “как”

Все знают или должны знать, что требования описывают, какая измеримая функциональность должна выполняться, а проект описывает, как эта функциональность реализуется. Но сравните проектировщика и реализатора: проектировщики улучшают свои решения через прототипирование, а прототипирование уже является реализацией. Когда реализатор меняет реализацию, он часто меняет и проект. В этом и состоит противоречие: граница между требованиями и проектом действительно размыта, а такая размытость означает субъективность и открывает дверь допущениям. В авиации это особенно заметно: когда требование к воздушному судну является системным требованием, когда требованием к ПО или аппаратуре, а когда проектным решением? Фух. Продолжим.

Для авиационного ПО, как бортового, так и наземного, соответствующие руководящие документы по сертификации DO-178C и DO-278A требуют, чтобы разработка требований высокого уровня предшествовала разработке требований низкого уровня. Требования высокого уровня всегда должны быть свободны от проектных деталей. Требования низкого уровня, напротив, находятся рядом с проектированием и считаются частью

процесса проектирования ПО, который включает и архитектуру, и требования низкого уровня. Общая последовательность этапов авиационной программной инженерии показана ниже:



Последовательность этапов авиационной программной инженерии

Создание хороших требований и стандартов требований

Вы когда-нибудь наблюдали за профессиональными игроками в бильярд? Случайному зрителю они кажутся не только умелыми, но и часто удачливыми. Они не просто редко промахиваются: биток остается в идеальной позиции для следующего удара. Хорошие любители забивают шар, а профессионалы планируют несколько ударов вперед и выигрывают партию серией. В этом разница между профессионалами и любителями. В авиации удача встречается редко, за исключением невезения, о котором на следующий день пишут в новостях о катастрофе. Большинство авиационных происшествий вызвано не пилотами, а неспособностью пилотов снизить последствия. Впрочем, это отдельная тема. Как профессионалы планируют несколько ходов вперед, так и авиационные инженеры должны делать то же самое при разработке требований. Авиационные требования должны исходить из желаемого конечного результата. Вместо любительской

ошибки, когда текстовые требования придумывают по одному, как отдельные удары в любительском бильярде, профессионал должен заранее видеть итоговый результат: что именно составляет успешную авиационную систему со всеми ее взаимодействиями.

Авиационные руководящие документы дают на удивление мало информации о том, как разрабатывать требования, потому что такие методы весьма субъективны и потому трудно оцениваемы. Документы вроде DO-178C для бортового ПО и DO-278A для наземных/спутниковых систем и ПО в области CNS/ATM вместо этого ссылаются на “цели”, включая цели оценки требований. Они называются целями по простой причине: критерии оценки целей можно применять однозначно. Но у хороших требований есть и субъективные критерии, которые авиационные руководящие документы не предписывают и даже не рассматривают. Эти критерии субъективны, а потому не оцениваются через общие цели документов. Как же сертифицирующие авиационные органы обеспечивают включение таких субъективных критериев оценки требований? Очень просто: для этого и нужен обязательный стандарт требований, предписанный для DO-178C, DO-254 и DO-278A и настоятельно рекомендуемый для ARP4754A. Авиационные руководящие документы требуют, чтобы разработчик создавал и применял обязательный стандарт требований для более высоких уровней гарантии разработки, то есть для тех уровней, где отказ системы может привести к травмам или более тяжелым последствиям.

Теперь вернемся к итоговой цели: разработке успешных авиационных систем. Хотя авиационные руководящие документы по сертификации не требуют конкретной методологии или подхода к разработке требований, разработчик обязан доказать соответствие. Поэтому эти документы “требуют”, чтобы до начала разработки требований существовали следующие артефакты:

Необходимые входные артефакты для начала разработки авиационных требований в соответствии с руководящими документами

1. **План разработки.** Описывает стратегию разработки требований.
2. **Стандарт требований.** Задаёт критерии составления и оценки требований.
3. **Контрольный перечень рассмотрения требований.** Используется при формальном рассмотрении требований.
4. **Матрица трассируемости.** Показывает двунаправленные связи между уровнями требований, кодом и тестированием.

Четыре входа процесса разработки авиационных требований, показанные выше, должны быть определены до начала разработки требований. Последующая оценка требований должна использовать все четыре входа, чтобы подтвердить, что применялись процессы, соответствующие руководящим документам. Что отличает любителей от профессионалов в игре под названием “разработка требований”? Эффективность, точность и качество:

- **Эффективность:** прояснить требования рано и с минимальным числом итераций;

- **Точность:** обеспечить корректность и полноту требований;
- **Качество:** обеспечить соответствие требований, выполнение всех критериев перехода, то есть входных критериев, и наличие записей рассмотрений и аудитов.

Чтобы добиться эффективности, точности и качества при разработке требований, нужно мыслить как профессионалы и сначала видеть конечный результат. Для любителя работа с авиационными требованиями сводится к тому, чтобы удовлетворить ожидания заказчика, реализовав его исходные запросы. Для профессионалов конечная цель успешной авиационной системы шире: соответствовать всем руководящим документам по сертификации, удовлетворять требованиям, связанным с безопасностью, затем функциональным требованиям, включая требования, пришедшие от заказчика, корректно взаимодействовать с другими системами и поддерживать повторное использование. Иными словами, профессиональный разработчик требований планирует на несколько ходов вперед и учитывает намного больше, чем ближайшую задачу: просто реализовать исходные требования заказчика. Каковы же атрибуты профессиональных авиационных требований? Следующий рисунок дает краткое резюме. Обратите внимание: нужно прилагать все усилия для повторного использования существующих технологий. Авиация и так сложна; делать ее еще сложнее необязательно.

Атрибуты хороших авиационных требований

- **Корректные и полные.** Точно и в полном объеме описывают функциональность, подлежащую реализации.
- **Реализуемые.** Возможна реализация с учетом известных возможностей и ограничений системы и ее среды.
- **Необходимые.** Определяют необходимость либо требуются для соответствия внешнему требованию, интерфейсу или стандарту.
- **Приоритетные.** Являются существенными; связь с другими требованиями задана через приоритет относительно этих требований.
- **Однозначные.** Допускают только одно толкование требования одним или несколькими разработчиками и проверяющими.
- **Верифицируемые.** Позволяют оценить или испытать реализацию для подтверждения ее корректности.

Слабые требования

Слабые требования встречаются повсюду, но обычно не из-за предполагаемой некомпетентности или недостатка знаний у инженеров. Чаще причина противоположная: умные инженеры считают, что требования пишутся для них самих или для таких же умных людей, либо формулируют между собой и руководителями/заказчиками слабо выраженные цели. В этом и проблема. Другой инженер, которому нужно читать требования, чтобы верифицировать заявленную функциональность или изменить ее в будущем, почти наверняка не обладает знаниями первоначального создателя или

реализатора. Если системы сравнительно малы или разрабатываются исключительно выдающимися инженерами, вспомните Национальное управление по аэронавтике и исследованию космического пространства (National Aeronautics and Space Administration, NASA) и раннюю космическую программу, авиационные требования, откровенно говоря, менее важны. По мере того как системы становятся крупнее и сложнее, требуется больше инженеров, и средний уровень квалификации приближается к буквальному смыслу слова “средний”. Именно системы таких инженеров больше всего страдают от слабых требований. По правде говоря, подавляющее большинство авиационной разработки строится как эволюционное развитие прежних воздушных судов и систем; полностью новые воздушные суда или системы встречаются редко. Поэтому первоначальные разработчики редко полноценно участвуют в производных версиях, а слабые требования теряют необходимые знания этих создателей.

Рассмотрим следующее очевидно слабое требование:

“Система должна, в качестве цели, вычислять курс воздушного судна, высоту и глобальное положение с достаточной точностью.”

Очевидно, приведенное требование слабое. Но если разобрать его слово за словом, станет ясно, что оно на самом деле *очень* слабое.

- ***“Система”***: недостаточно. Какая система? Какая версия и конфигурация?
- ***“в качестве цели”***: недостаточно. Цели не являются требованиями и не поддаются оценке.
- ***“вычислять”***: недостаточно. Вычислять как? По какому алгоритму? Когда? Как часто?
- ***“курс воздушного судна”***: недостаточно. Истинный курс или магнитный? Какая точность?
- ***“высоту”***: недостаточно. Над уровнем моря или над местностью? В футах или метрах? Какое разрешение?
- ***“и глобальное положение”***: недостаточно. Широта/долгота или Всемирная система географических координат (World Geographic Reference System, GEOREF)? Какое разрешение?
- ***“с достаточной точностью”***: недостаточно. Допуск совершенно неясен.

На самом деле в этом требовании есть только одно “хорошее” слово: ***“должна”***. Оставьте его и исправьте все остальное.

Требования, цели и субъективные пожелания

Распространенная проблема критичных для безопасности систем состоит в том, что системные цели формулируются как требования. Цели, как новогодние обещания, прекрасны, и всем было бы полезно их иметь. Но цели нельзя объективно оценить и гарантировать. Ниже приведены распространенные авиационные цели, которые слишком часто записывают в формате требований:

“Воздушное судно должно быть безопасным.”

“Серьезные происшествия не могут происходить.”

“Пассажиры и экипаж никогда не могут быть убиты или травмированы.”

Все это действительно великие цели, но это не требования. Еще одна частая проблема требований – выражение через субъективные пожелания. Люди насыщали сообщения субъективными пожеланиями с начала письменной речи. Расстояния, например “далеко” и “коротко”, размер, например “большой” и “малый”, температура, например “горячий” и “холодный”, фиксировались уже в ранних человеческих языках. Но что именно они означают? Читатель может представить общий смысл, но не получает ничего дискретно верифицируемого. Рассмотрим следующие субъективные слова:

- Немедленно
- Реагировать
- Краткий
- Одновременный
- Модульный
- Номинальный

Такие слова часто встречаются в требованиях к критичным для безопасности системам, но их нужно заменять объективными терминами. Следующие требования взяты из спецификаций авиационных требований, которые автор получал в прежней жизни, будучи сооснователем и генеральным директором крупнейшей в мире компании по услугам авиационной сертификации. Это не объективные требования и, по сути, едва ли даже слабые цели:

- ПО должно быть робастным.
- Аппаратура должна плавно деградировать под нагрузкой.
- ПО должно разрабатываться в соответствии с современными практиками программирования.
- Система должна обеспечивать необходимую обработку во всех режимах работы.
- Использование памяти вычислителя должно быть минимизировано для учета будущего роста.
- ПО и аппаратура должны быть удобны в использовании.

Ясно, что разработка великих требований требует уникального сочетания науки, искусства, навыка, практики и знания общей функциональности и возможностей системы. Не обещайте Луну никому, кроме своей невесты. Но еще нужны хорошие планы, стандарты, контрольные перечни и методологии. Следующие лучшие практики не требуются авиационными руководящими документами, потому что их нельзя объективно оценить, но они должны быть встроены в жизненный цикл инженерии требований:

Лучшие практики разработки требований

- Включать примеры требований в стандарт требований.
- Рассматривать требования небольшими пакетами связанной функциональности.
- Рассматривать требования в контексте полной двунаправленной трассируемости, используя все связанные отношения.
- Включать подробности обработки ошибок, допусков, встроенного контроля и сброса: это типичные области слабых требований в авиации.
- Использовать подробные профессиональные контрольные перечни для рассмотрения требований.
- Разрабатывать тестовые примеры на основе требований в рамках рассмотрения; улучшать требования по выявленным недостаткам тестовых примеров.

Хотя эти лучшие практики не являются обязательными, любой, кто разрабатывает авиационные требования без их учета, надеется на удачу. Любители надеются на удачу; профессионалы планируют успех. Мы все можем быть профессионалами.

DO-200B: аэронавигационные данные

Автор: Вэнс Хилдерман

Рецензенты:

- Г-н Дон Коулман, представитель Федерального управления гражданской авиации США по инженерной экспертизе (FAA Designated Engineering Representative, FAA DER); г-н Том Рот, представитель Федерального управления гражданской авиации США по инженерной экспертизе

DO-200B, “**Стандарты обработки аэронавигационных данных**” (Standards for Processing Aeronautical Data), – один из краеугольных камней современной авиации. Хотя DO-178 и DO-278 обычно получают больше внимания, именно DO-200B служит рабочей основой, на которую прямо или косвенно опираются все современные воздушные суда. Почему? Потому что DO-200B определяет, как подготавливаются, обновляются, используются и сопровождаются данные, необходимые для безопасной эксплуатации воздушных судов. Показательно, что DO-178 и DO-278 прямо названы руководствами: в названиях этих документов используется слово Considerations (“соображения”), а не Standard (“стандарт”). В отличие от них DO-200B действительно является минимальным стандартом, что прямо следует из его названия.

Рассмотрите следующие утверждения и определите, истинны они или ложны. Ответы и пояснения приведены далее в этой главе:

1. **И / Л:** DO-200B применяется к данным о рельефе, навигационным данным и данным двигателя.
2. **И / Л:** DO-200B – существенно переработанная версия DO-200A.
3. **И / Л:** Три базовых процесса DO-200B – это требования к качеству данных (Data Quality Requirements, DQR), требования к обработке данных и система менеджмента качества.
4. **И / Л:** Поставщик несет основную ответственность за обеспечение целостности использования данных.
5. **И / Л:** Письмо о приемке типа 1 (Type 1 Letter of Acceptance, Type 1 LOA) требует испытаний на указанном конечном изделии авионики.
6. **И / Л:** Все инструменты обработки данных должны быть квалифицированы для данных уровня гарантии 1.

Если эти вопросы действительно показались вам простыми, можно поздравить себя с подлинной экспертностью. Если же они только выглядели простыми, последующий материал этой главы как раз для вас. Отвечать на эти вопросы без понимания общей

цепочки данных и роли гарантии качества в DO-200B – все равно что понимать преобразования Фурье, не понимая математический анализ. Для простых смертных это невозможно...

В самом кратком виде DO-200B дает руководство по следующим аэронавигационным аспектам:

- минимальные стандарты и руководство по обработке аэронавигационных данных;
- “аэронавигационные данные” – это данные, используемые для навигации, планирования полета, предупреждения столкновения с землей, летных тренажеров и т. д.;
- критерии разработки, изменения и сопровождения аэронавигационных данных;
- в конечном счете – предоставление пользователю гарантии качества данных.

Значение термина “аэронавигационный”

Сначала разберемся, что означает термин “аэронавигационный”. Он относится не только к воздушным судам, управлению воздушным движением, учебным средствам и навигации. Термин выбран именно потому, что он шире понятия “воздушное судно”. Если DO-178 и DO-254 предназначены для бортового ПО, то DO-200B применяется к данным, которые могут никогда не попасть на борт воздушного судна, но тем или иным образом влияют на безопасность полетов. Сюда входят эксплуатация воздушных судов, моделирование, обучение, планирование и другие области. Краткий обзор DO-200B приведен ниже:

Краткий обзор DO-200B

- Документ Радиотехнической комиссии по аэронавтике (Radio Technical Commission for Aeronautics, RTCA) DO-200B: "Стандарты обработки аэронавигационных данных".
- Впервые разработан в 1988 году; затем в 1995-1998 годах специальным комитетом RTCA 181 (Special Committee 181, SC-181); затем снова для версии 2015 года: DO-200B.
- Разработан в основном промышленностью, при участии части государственных специалистов.
- Содержит много компромиссов для согласования разных целей.
- Не является набором рецептов и не дает точного руководства "как сделать".
- Имеет "дискуссионную" структуру руководства; допускает множество разных разработок и типов данных.
- На практике: золотое правило для аэронавигационных данных.

Минимальные стандарты

DO-200B представляет собой умеренное обновление DO-200A: в нем усилен акцент на аэронавигационных данных в целом (DO-200A была больше ориентирована на навигацию), на информационной безопасности данных, на цепочке аэронавигационных данных, на расширенной области применения и определении квалификации инструментов с учетом DO-330, а также на более ясных определениях. DO-200B задает именно “минимальные” стандарты. Пользователю рекомендуется, а часто и необходимо, делать больше, чем предусмотрено этим минимумом. Поскольку аэронавигационные данные могут иметь множество разных форм, а будущее авиации наверняка принесет формы данных, которые сегодня еще неизвестны, DO-200B не может содержать достаточно подробных указаний для каждого типа данных. Аналогичная ситуация есть и с ПО и аппаратурой в DO-178 и DO-254: эти стандарты применимы практически ко всей бортовой авионике, от освещения в туалете до устройств реверса тяги и навигационных систем, но дополнительные требования для каждого конкретного типа системы в них не рассматриваются. Для бортовой авионики такие дополнительные системно-специфические требования содержатся в других обязательных сертификационных документах, например в технических стандартных приказах (Technical Standard Order, TSO); подробности см. в предыдущей главе. В отличие от бортовых систем, дополнительные требования к конкретным видам аэронавигационных данных в значительной степени не покрыты вспомогательными документами. Заметное исключение – Консультативный циркуляр (Advisory Circular, AC) 20-153B, **“Принятие процессов обработки аэронавигационных данных и связанных баз данных”** (Acceptance of Aeronautical Data Processes and Associated Databases). Он находится в свободном доступе и обязателен к прочтению для всех, кто работает с DO-200B. Поэтому пользователям DO-200B важно помнить: “минимальные” стандарты почти наверняка недостаточны для большинства проектов. Пользователь должен дополнительно определить критерии, относящиеся именно к его данным и процессам. Следующий рисунок кратко показывает ключевые различия между DO-200A и DO-200B:

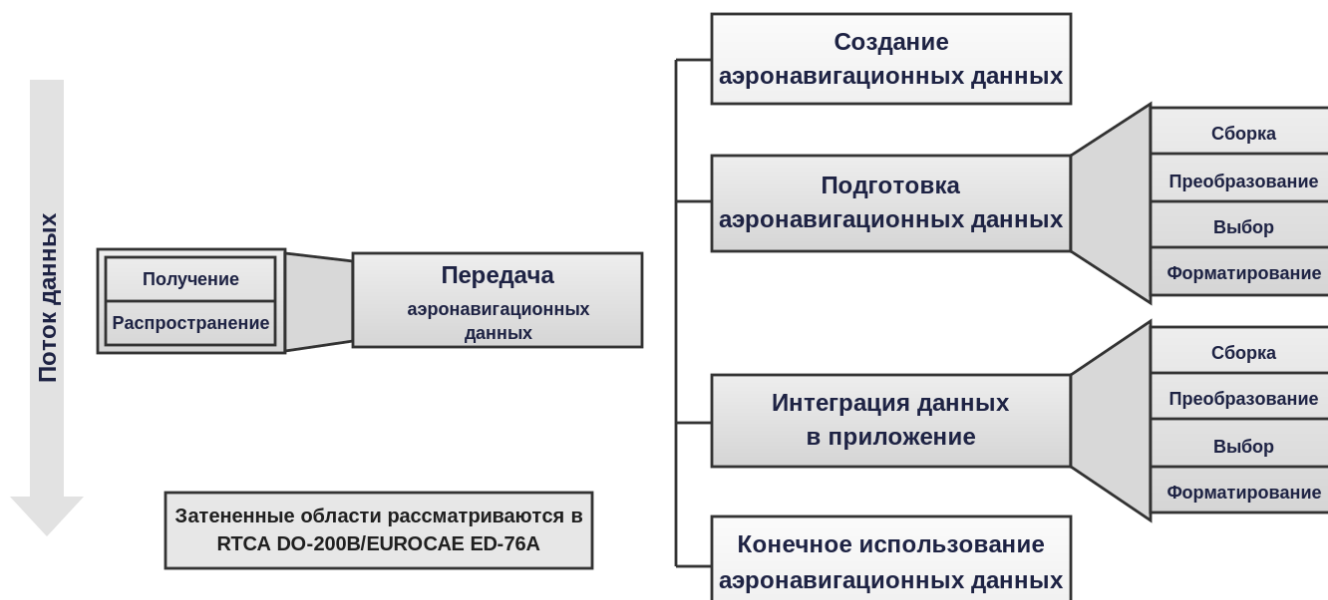
Ключевые различия DO-200A и DO-200B

- Расширенная область: все аэронавигационные данные, не только навигационные.
- Применяется DO-330 для квалификации инструментов, то есть программных инструментов обработки данных.
- Уточнены конкретные документы, необходимые для соответствия DO-200B.
- Уточнены "процессы", "этапы" и "функции".
- Обязательная система менеджмента качества (Quality Management System, QMS), а также процесс оценки безопасности.
- Более строгая квалификация инструментов с применением DO-330.
- Более строгие цели и подробные требования к качеству данных (Data Quality Requirements, DQR).

DO-200B задает “рекомендуемые стандарты”, а не жесткие требования. Документ разрабатывали более 245 специалистов со всего мира, главным образом из компаний, занимающихся разработкой авиационной продукции, а также представители ключевых сертифицирующих органов и военные специалисты. Как почти в любом стандарте, созданном комитетом, авторам DO-200B пришлось согласовывать противоречивые интересы и весьма распространенное желание написать невозможный идеальный документ, то есть сделать его “всем для всех”. Поэтому 77 страниц DO-200B охватывают большие и малые наборы аэронавигационных данных, разные уровни критичности и разных пользователей. Главная идея в том, что для аэронавигационных данных доказуемые системы качества важнее жестко предписанных шагов процесса. Поэтому каждый пользователь должен внимательно рассмотреть и затем проанализировать свой вклад в безопасность, задавая для каждого шага своей цепочки данных следующие вопросы:

- *Может ли использование данных на этом шаге не обнаружить ошибку?*
- *Может ли использование данных на этом шаге внести ошибку?*
- *Может ли использование данных на этом шаге распространить ошибку?*
- *Каковы ответы на эти вопросы с учетом экосистемы разработки и применения данных, а также цепочки инструментов?*

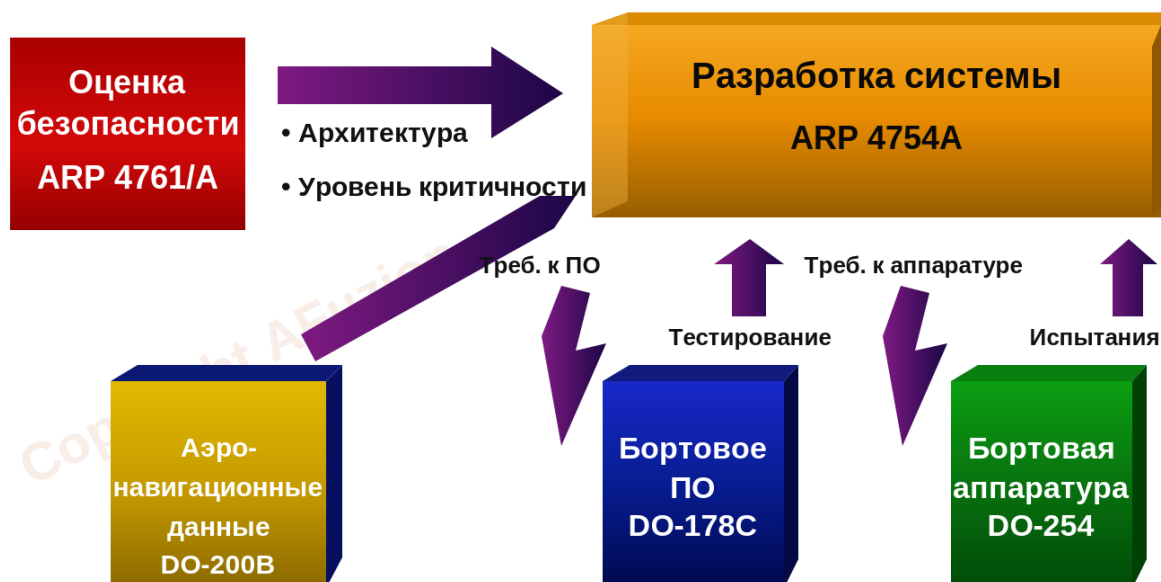
Область действия DO-200B и процесс цепочки аэронавигационных данных показаны на следующем рисунке:



Затененные блоки, рассматриваемые в DO-200B/EUROCAE ED-78

Аэронавигационные данные и авиационная экосистема

DO-200B входит в общую экосистему авиационной разработки (Aviation Development Ecosystem) – термин, предложенный AFuzion. Эта экосистема включает бортовое и наземное ПО и системы, а также формализованные процессы оценки безопасности. Следующий рисунок показывает эти взаимосвязи:



Четыре основные цели DO-200B кратко показаны на следующем рисунке:



Четыре основные цели DO-200B

DO-200B требует планирования, определения требований к данным, требований к обработке и свидетельств выполнения связанных процессов. Затем все это должно быть валидировано и верифицировано по всей цепочке данных, от получения данных до их передачи. И поставщик, и пользователь совместно отвечают за обеспечение целостности данных. Ниже показаны документы, обычно требуемые для типовой деятельности по DO-200B:

Обычно требуемые документы деятельности по DO-200B

- План соответствия.
- Документ требований к качеству данных (Data Quality Requirements, DQR).
- Документ процедур обработки данных.
- Документы квалификации инструментов.
- Документы управления конфигурацией: план и записи.
- Документ управления компетенциями.
- Документ системы менеджмента качества (Quality Management System, QMS).

Основной план DO-200B – план соответствия (Compliance Plan). Этот план охватывает темы, показанные на следующем рисунке:

Ключевые темы плана соответствия DO-200B

1. Определение требований к качеству данных (Data Quality Requirements, DQR).
2. Определение требований к обработке аэронавигационных данных.
3. Определение требований к системе менеджмента качества, СМК (Quality Management System, QMS).
4. Определение персонала, ответственного за соответствие требованиям.
5. Заявление об используемых стандартах.
6. Заполненная матрица соответствия.

Определения аэронавигационных данных

В авиационной сфере уже есть собственный словарь. Ранее в книге уже вводилось жаргонное обозначение Avilish – “авиационный английский”. DO-200B идет еще дальше и вводит ряд ключевых терминов, которые вне области аэронавигационных данных почти не имеют понятного контекста:

Ключевые термины DO-200B, часть 1

- **Аэронавигационные данные:** данные, используемые для аэронавигационных применений, таких как навигация, планирование полета, летные тренажеры, предупреждение столкновения с землей и другие цели; включают навигационные данные, данные о рельефе и препятствиях.
- **Аэронавигационная база данных:** данные, хранящиеся в электронном виде в системе, которая поддерживает бортовые или наземные аэронавигационные применения. Аэронавигационная база данных может обновляться через регулярные интервалы.
- **Сборка:** процесс объединения или компиляции аэронавигационных данных, иногда поступающих от нескольких поставщиков данных, в базу данных и установления базовой версии для последующей обработки. Этап сборки включает проверку данных и обеспечение устранения выявленных ошибок и пропусков.
- **Уровень гарантии:** степень уверенности в том, что элемент данных не искажен при хранении или передаче. Может классифицироваться по трем уровням: 1, 2 и 3; при этом 1 соответствует наивысшей степени уверенности, а 3 – наименьшей.

Ключевые термины DO-200B, часть 2

- **Корректные данные:** данные, удовлетворяющие установленным требованиям к качеству.
- **Качество данных:** степень или уровень уверенности в том, что предоставленные данные удовлетворяют требованиям пользователя. Эти требования включают точность, разрешающую способность, уровень гарантии, трассируемость, своевременность, полноту и формат.
- **Конечный пользователь:** последний пользователь в цепочке аэронавигационных данных. Конечными пользователями аэронавигационных данных обычно являются эксплуатанты воздушных судов, подразделения планирования авиакомпаний, поставщики обслуживания воздушного движения, поставщики летных тренажеров, изготовители планеров, системные интеграторы и регулирующие органы. Обычно отвечает за качество данных и соответствие DO-200B.
- **Ошибка:** дефектные или деградировавшие элементы данных либо потерянные или неверно размещенные элементы данных, либо элементы данных, не удовлетворяющие установленным требованиям к качеству.
- **Первичный источник:** первая организация в цепочке аэронавигационных данных, принимающая ответственность за данные. Например, государство или организация, соответствующая документам Радиотехнической комиссии по аэронавтике (Radio Technical Commission for Aeronautics, RTCA) DO-200B и Европейской организации по оборудованию для гражданской авиации (European Organisation for Civil Aviation Equipment, EUROCAE) ED-76.

Атрибуты качества аэронавигационных данных

Аэронавигационные данные имеют следующие атрибуты качества, как правило применимые к каждому элементу данных или набору данных:

Атрибуты качества аэронавигационных данных

Необходимо установить характеристики качества данных для предполагаемого использования.

Характеристики должны быть зафиксированы и должны помогать формировать требования к качеству данных в части:

- точности;
- разрешающей способности;
- уровня гарантии;
- трассируемости;
- своевременности;
- полноты;
- формата.

Атрибуты качества данных кратко показаны на следующем рисунке:

Сводка атрибутов качества данных

1. Точность данных.
2. Разрешающая способность данных.
3. Доказуемое предотвращение искажения в цепочке аэронавигационных данных.
4. Трассируемость – способность определить происхождение и исправить дефекты.
5. Действительность периода применения, например "окна использования".
6. Полнота данных: есть все необходимое.
7. Формат: формат поставщика соответствует потребностям следующего пользователя; выход каждого звена соответствует входу следующего звена.

Точность данных и связь с DO-201

Перед применением DO-200B важно понять применимые требования к данным и происхождение этих данных. Для этого предназначен DO-201, **“Отраслевые требования к аэронавигационной информации”** (Industry Requirements for Aeronautical Information). DO-201 обеспечивает количественное задание точности данных, разрешающей способности, критичности, расчетов и процедур. Когда это применимо, требования к точности данных должны быть явно заданы. Например, рассмотрим всенаправленный азимутальный радиомаяк (VHF Omnidirectional Range, VOR). У каждого VOR есть уникальный идентификатор. Этот идентификатор – просто “имя”, и оно должно быть точным, то есть соответствовать фактическому VOR с тем же именем. Для самого имени

требований к точности нет. Однако у того же VOR есть местоположение и высота, а для них существуют соответствующие требования к точности. Именно эти требования должны быть известны, измеримы и оценены.

Разрешающая способность данных

Аэронавигационные данные часто имеют связанные атрибуты разрешающей способности. Требуемые характеристики должны быть заданы исходя из предполагаемого использования этих данных. Нередко разные подготовители аэронавигационных данных не до конца понимают, как именно данные будут использоваться, поэтому предполагаемое использование обязательно должно быть документировано и известно. Разрешающая способность относится только к измеримым элементам. Например, в приведенном выше примере имя VOR служит идентификатором и не имеет атрибута разрешающей способности, а местоположение и высота измеримы, поэтому для них должны быть документированы соответствующие характеристики разрешающей способности. Согласно DO-201, разрешающая способность данных определяется и включается в формат выходных данных. Если конкретный элемент данных не охвачен DO-201, пользователь должен выполнить оценку безопасности, определить надлежащие характеристики разрешающей способности и задокументировать их.

Уровни гарантии и уровни критичности

В то время как DO-178, DO-254 и родственные стандарты используют связанные уровни критичности, DO-200B использует уровни гарантии (Assurance Levels) 1, 2 и 3. В таблице ниже также используется сокращение DPAL – уровень гарантии процесса обработки данных (Data Process Assurance Level, DPAL). Связь между уровнями гарантии и уровнями критичности для более крупных самолетов транспортной категории с неподвижным крылом по части 25 правил FAA (14 CFR Part 25) показана ниже:

Помните:

Уровень А $\leq 1E-09$	Уровень В $\leq 1E-07$	Уровень С $\leq 1E-05$	Уровень D $> 1E-05$	Уровень Е не применимо
----------------------------------	----------------------------------	----------------------------------	-------------------------------	----------------------------------

И:

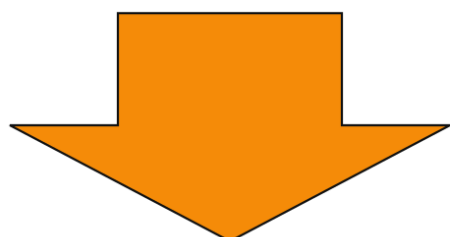
Таблица В-1. Уровни гарантии

Уровень гарантии процесса обработки данных	Связанное требование к данным государства (ИКАО)
1	Критичные
2	Существенные
3	Обычные

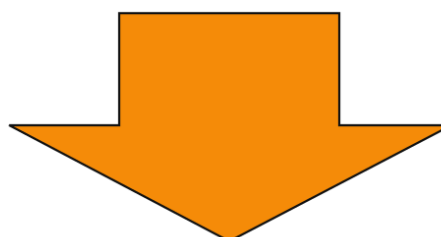
DPAL	Вероятность необнаруженной ошибки
1	$\leq 10^{-8}$
2	$\leq 10^{-5}$
3	Не применимо.

Три уровня гарантии DO-200B означают, что данные делятся на три категории; уровень 1 – наиболее строгий. Благодаря разным уровням гарантии для менее критичных элементов данных можно сократить отдельные сертификационные мероприятия, уменьшая время и бюджет, связанные с обработкой таких элементов. Для дополнительного представления об уровнях гарантии в DO-200B и их связи с безопасностью полетов рассмотрите следующую эквивалентность:

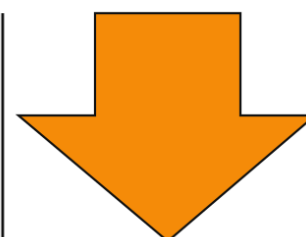
Уровень А $\leq 1E-09$	Уровень В $\leq 1E-07$	Уровень С $\leq 1E-05$	Уровень D $> 1E-05$	Уровень Е не применимо
----------------------------------	----------------------------------	----------------------------------	-------------------------------	----------------------------------



**Уровень
гарантии 1**

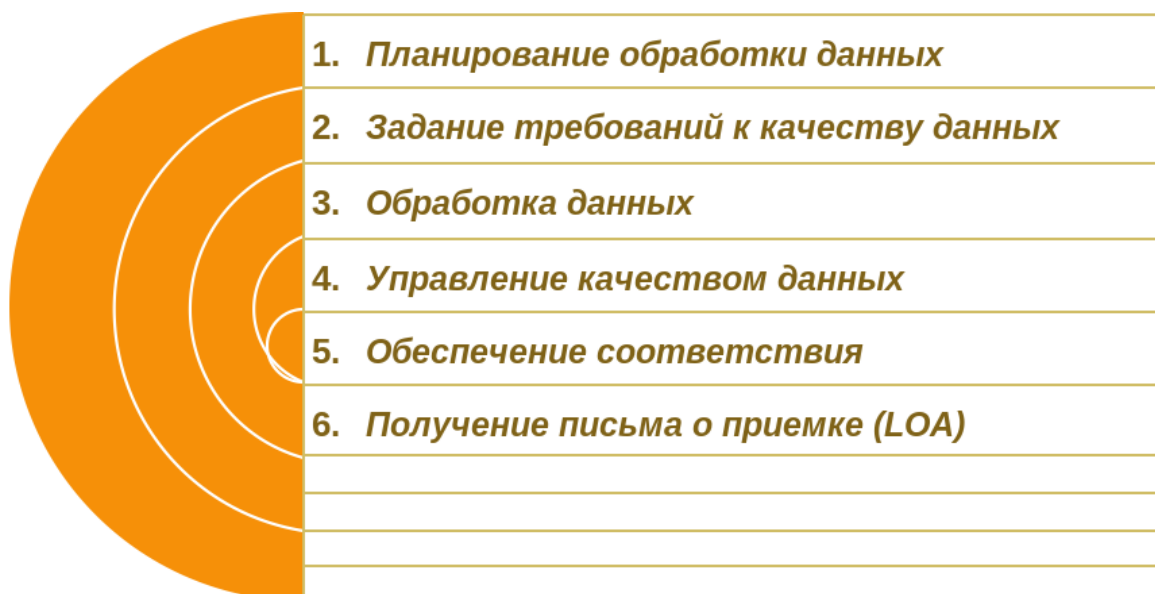


**Уровень
гарантии 2**



**Уровень
гарантии 3**

Процесс планирования соответствия DO-200B должен обеспечивать, чтобы ключевые действия по гарантии целостности данных были запланированы, выполнены, оценены и зафиксированы. Для достижения соответствия DO-200B в отношении данных следующие шесть интегральных действий должны применяться на протяжении всей разработки и развертывания данных:



Система менеджмента качества (QMS) DO-200B

Авиационные системы реального времени, включающие ПО и аппаратуру, тщательно разрабатываются с индивидуальной оценкой аспектов, связанных с безопасностью. Обработка аэронавигационных данных, напротив, почти всегда строится вокруг готовых коммерческих изделий (Commercial Off-The-Shelf, COTS): аппаратуры и ПО, используемых для обработки данных. Поскольку разработку таких готовых средств нельзя подвергнуть внутреннему контролю авиационного уровня, соответствие аэронавигационных данных сосредоточено не на самих готовых средствах, а на среде обработки данных, специально построенной для DO-200B с их использованием. Как и в большинстве высококачественных систем обработки данных, применяемых в финансовых учреждениях и на производстве, система менеджмента качества, СМК (Quality Management System, QMS) обязательна. DO-200B требует лишь небольших дополнений к уже хорошей СМК, поэтому любая доказуемо эффективная СМК должна быть достаточна для DO-200B после незначительных доработок. Хотя о коммерческих СМК написаны сотни статей и множество книг, для DO-200B применимо следующее краткое резюме:

Что такое система менеджмента качества (Quality Management System, QMS)?

- Организационное закрепление мероприятий, связанных с качеством, во всей организации.
- Внедрение практик качества для постановки целей, последующей постоянной оценки и улучшения.
- Внедрение формально определенных письменных процессов с обучением всех вовлеченных лиц для выполнения и мониторинга.

Следовательно, СМК, соответствующая DO-200B, необходима для последующего одобрения передаваемых данных. Большинство организаций, обрабатывающих авиационные данные, используют коммерческие СМК на основе стандарта ISO 9001 Международной организации по стандартизации (International Organization for Standardization, ISO), аэрокосмического стандарта AS9115A или интеграции моделей зрелости возможностей (Capability Maturity Model Integration, CMMI) уровня 4. Атрибуты хорошей СМК и, тем более, СМК, соответствующей DO-200B, перечислены ниже:

Атрибуты системы менеджмента качества, соответствующей DO-200B

- Возможность оценить реализацию относительно плана.
- Возможность доказать, что определенные письменные процедуры выполнялись так, как задано процессной базой.
- Оценка выполненной работы относительно письменной базы.
- Оценка возможностей рассмотрения, включая контрольные перечни для мероприятий.
- Пересмотр процедур, позволяющих доказать соответствие.
- Определение контрольных перечней аудита.
- Ответ на практический вопрос: можете ли вы доказать, что будете признаны "невиновными", а не виновными?
- Учет сходства с интеграцией моделей зрелости возможностей (Capability Maturity Model Integration, CMMI) уровней 3-4 и DO-178C уровня C; всегда с опорой на ISO 9001 согласно системе менеджмента качества (Quality Management System, QMS).

Пример уровня гарантии аэронавигационных данных

В качестве примера рассмотрим данные, связанные с точными заходами на посадку. Оценка безопасности показала, что доступность возможности точного захода на посадку соответствует состоянию менее тяжелому, чем опасное, но более тяжелому, чем незначительное:

- а. Катастрофическое
- b. Опасное
- c. **Серьезное**
- d. Незначительное
- e. Без последствий

Поскольку это состояние класса "серьезное", по DO-200B ему соответствует уровень гарантии 2.

Трассируемость данных

DO-200B требует, чтобы была определена возможность выполнять сквозную трассируемость данных и имелись свидетельства, подтверждающие эту возможность. Для данных трассируемость означает способность установить их происхождение. В большинстве случаев пользователи аэронавигационных данных не являются источниками данных, которые они используют, обрабатывают или распространяют. Должна существовать возможность определить источник любой ошибки в данных. Следовательно, данные должны быть прослеживаемы от любого поставщика к следующему пользователю, чтобы можно было установить коренную причину любой ошибки. Когда данные исправляются, последующие пользователи должны получать уведомления как об ошибках, так и о внесенных исправлениях.

Период действия, полнота и формат

Данные должны быть верифицированы с точки зрения их действительности в соответствующем периоде, полноты и формата. Если у данных, например о местоположении и ориентации ВПП, есть установленный период использования продолжительностью 28 дней, этот период должен быть задан, а действительность данных должна определяться с его учетом. Атрибуты полноты данных также должны быть заданы. Например, если данные предназначены только для больших воздушных судов, ВПП короче 3000 футов можно не включать, и при отсутствии таких коротких ВПП набор данных все равно может считаться полным. Формат данных должен быть задан как для определения действительности данных, так и для определения их использования. При этом нужно учитывать следующего пользователя в цепочке аэронавигационных данных. Например, если данные о превышении аэродрома состоят из двух элементов: 1) превышение и 2) признак “выше” или “ниже” уровня моря, то эта связь должна быть определена, а набор данных, то есть превышение вместе с индикатором выше/ниже уровня моря, должен быть валидирован соответствующим образом.

Пример применения DO-200B

Обычно исходная точка для аэронавигационных данных – поставщик аэронавигационных данных, который уже квалифицировал свои данные с использованием DO-201, “Отраслевые требования к аэронавигационной информации”. Соответствующая DO-201 аэронавигационная база данных такого поставщика обычно включает все данные, доступные на аэронавигационных картах. Затем DO-200B используется для извлечения именно тех данных, которые требуются конкретному изделию для выполнения его предполагаемой функции в полете. Это лишь небольшое подмножество данных, предоставленных поставщиком, но оно должно удовлетворять рассмотренным выше атрибутам качества аэронавигационных данных.

Данные из аэронавигационной базы данных поставщика обычно распространяются каждые 28 дней, то есть 13 циклами в год. Значительная часть этих данных содержит даты применимости, позволяющие вносить изменения внутри 28-дневного цикла. Затем приложение DO-200B каждые 28 дней извлекает данные из гораздо более крупной аэронавигационной базы данных и передает их подписчикам для использования в поставляемых ими изделиях. DO-200B содержит руководство как по квалификации инструмента извлечения данных, так и по верификации, которую необходимо выполнять при каждом извлечении.

Для достижения этой двойной цели большинство поставщиков авиационных систем квалифицируют инструмент извлечения данных только при изменении самого инструмента. Инструмент извлечения данных сам выполняет шаги верификации, требуемые при каждом извлечении в 28-дневном цикле. Поэтому артефакты, относящиеся к обновлениям инструмента извлечения данных, хранятся отдельно от данных, сохраняемых при каждом запуске инструмента, обычно раз в 28 дней. Данные, сохраняемые каждые 28 дней в результате выполнения этого инструмента, должны затем удовлетворять указанным выше атрибутам качества аэронавигационных данных.

DO-278A: наземные и спутниковые системы и ПО связи, навигации, наблюдения и организации воздушного движения (CNS/ATM)

Автор: Вэнс Хилдерман

Рецензент:

г-н Том Рот, представитель Федерального управления гражданской авиации США (Federal Aviation Administration, FAA) по инженерной экспертизе (Designated Engineering Representative, DER) по DO-278A: введение.

DO-278A часто называют “младшим братом DO-178 для наземных систем”. Но действительно ли DO-278A – младший брат, или это скорее старшая сестра? И относится ли он только к наземным системам? Ответы могут удивить...

Полное название DO-278A: *Руководство по обеспечению целостности ПО систем связи, навигации, наблюдения и организации воздушного движения (Communication, Navigation, Surveillance, and Air Traffic Management, CNS/ATM)*. Ключевой термин здесь – CNS/ATM, то есть авиационное ПО наземного базирования, связанное со связью, навигацией, наблюдением и организацией воздушного движения. Хотя большинство таких систем расположено на земле, некоторые из них являются бортовыми: например, космические системы наблюдения и самолеты дальнего радиолокационного обнаружения и разведки. Так, новейшие спутниковые системы связи Iridium, над которыми автор и 60 его инженеров работали во второй половине 1990-х годов, используют новое поколение спутников со встроенным оборудованием автоматического зависимого наблюдения-вещания, АЗН-В (Automatic Dependent Surveillance-Broadcast, ADS-B), для отслеживания воздушных судов; это спутниковое оборудование АЗН-В сертифицировано по DO-278A. Автор также участвовал в этой работе, но уже двумя десятилетиями позже.

При таком длинном названии ясно, что DO-278A – вовсе не просто младший брат DO-178C. DO-278 был обновлен до DO-278A специальным комитетом 205 (Special Committee 205, SC-205) и выпущен в декабре 2011 года. Как знает любой настоящий знаток истории, важна не только дата события, но и контекст того, что в это же время происходило в других местах. В случае SC-205 важно понимать, что параллельно обновлялся DO-178B, из которого появился DO-178C для авионического ПО; оба документа вышли почти одновременно. Более 95 % DO-278A совпадает с DO-178C, и причина веская: и системы CNS/ATM по DO-278A, и авионика по DO-178C могут влиять на безопасность полетов на разных уровнях критичности, а методы программной инженерии у них очень похожи. Проще говоря, изобретать велосипед было незачем.

Поскольку эта книга посвящена “экосистеме” разработки авиационных систем, нет смысла повторять сведения о DO-178C: они уже изложены в двух специальных главах, а также в главах по квалификации инструментов, разработке на основе моделей, объектно-ориентированным технологиям, гарантии качества ПО, переходам ПО, ошибкам и затратам. Вместо этого данная глава рассматривает основы DO-278A и ключевые отличия от DO-178C.

Как и бортовое ПО, то есть ПО, которое выполняется на борту воздушного судна или непосредственно влияет на исполнение такого ПО, системы CNS/ATM могут влиять на безопасность полетов. Многие аспекты связи, навигации, наблюдения и организации воздушного движения воздействуют на безопасность: одна ошибка может иметь тяжелые последствия. Поэтому к системам CNS/ATM важно применять процесс со следующими доказуемыми свойствами:

Вот он: DO-278A

- Оценивает потенциальное влияние на безопасность.
- Определяет влияние на безопасность и оценивает критичность.
- Предписывает инженерные процессы разработки в зависимости от влияния на безопасность.
- Максимально сохраняет сходство с другими авиационными стандартами, связанными с безопасностью, например DO-178C.
- Учитывает уникальные аспекты наземных систем: готовые коммерческие изделия (Commercial Off-The-Shelf, COTS), включая аппаратуру и ПО.
- Подлежит надзору международных и местных сертифицирующих органов.

Вот он: DO-278A

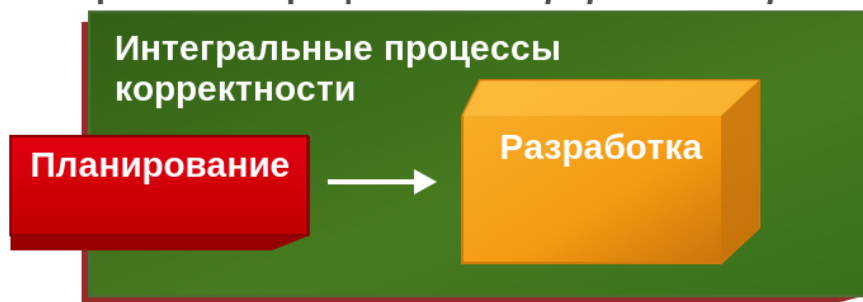
Так что же такое DO-278A? DO-278A – это вторая версия базового документа DO-278. Это родственный DO-178C документ, который, как отмечено выше, задает сходные рекомендации для безопасности бортового ПО: обычно такого ПО, которое выполняется на борту воздушного судна и участвует в обеспечении безопасности полета. Если вы уже понимаете DO-178C, то фактически знаете 95 % DO-278A: документы похожи, а многие аспекты идентичны. Это сходство позволяет опираться на гораздо более обширную литературу по DO-178C, поскольку публикаций по DO-278A сравнительно мало из-за меньшего числа изготовителей оборудования CNS/ATM. Но у знакомства с DO-178C есть и обратная сторона: человеческая природа склонна “проглядывать” тонкие различия, а это приводит к серьезным недопониманиям и ошибкам при применении DO-278A. Дальше глава одновременно вводит DO-278A для начинающих и показывает его отличия от DO-178C.

DO-278A – это содержательный руководящий документ, включающий рекомендации и цели, подлежащие оценке. Он предназначен для разработки наземных или спутниковых систем реального времени, содержащих ПО и связанных с эксплуатацией воздушных судов. Такие системы почти всегда широко используют готовые коммерческие изделия и технологии (Commercial Off-The-Shelf, COTS), включая аппаратуру и ПО. Наземные системы, на которые распространяется DO-278A, часто имеют значительно более крупные и разнообразные программные компоненты, чем их бортовые авионические аналоги. Кроме того, системы CNS/ATM эволюционировали из более ранних систем, появившихся до требования следовать DO-278, и по-прежнему содержат большие массивы унаследованного программного кода. Поэтому размер, разнообразие и повышенная зависимость от готовых коммерческих технологий играют ключевую роль в необходимости DO-278A; именно эти характеристики лежат в основе различий между DO-278A и DO-178C.

DO-278A и DO-178C: сходства

Подобно DO-178C, DO-278A опирается на процесс оценки безопасности для определения критичности системы/ПО. В бортовой области используется термин “уровень гарантии разработки” (Development Assurance Level, DAL), а для CNS/ATM он сокращается до “уровня гарантии” (Assurance Level, AL). В DO-278A есть три ключевых процесса: планирование, разработка и интегральное обеспечение корректности, как показано ниже:

- ▶ Процесс планирования – *выполняется первым*
- ▶ Процесс разработки – *следует за планированием*
- ▶ Интегральные процессы – *непрерывно в проекте*



Три ключевых процесса DO-278A

Ключевые темы DO-278A, включая указанные выше процессы планирования, разработки и интегрального обеспечения корректности, перечислены ниже:

Ключевые темы DO-278А в сопоставлении с тремя ключевыми процессами

Ключевые темы DO-278А:

1. Обзор.
2. Аспекты системы.
3. Жизненный цикл.
4. Процесс планирования.
5. Процесс разработки.
6. Валидация и верификация (Validation and Verification, V&V).
7. Управление конфигурацией.
8. Гарантия качества.
9. Одобрение.
10. Обзор систем связи, навигации, наблюдения и организации воздушного движения (Communication, Navigation, Surveillance / Air Traffic Management, CNS/ATM).
11. Данные и соображения.
12. Дополнительные соображения.

Три ключевых процесса:

1. Планирование.
2. Разработка.
3. Интегральные процессы корректности.

Ключевые темы DO-278А в сопоставлении с тремя ключевыми процессами

Как отмечалось ранее, DO-278А примерно на 95 % схож с DO-178С, особенно в следующих областях:

- *Оценка безопасности при постоянном участии специалистов по безопасности*
- *5 планов, 3 стандарта и связанные контрольные перечни*
- *Требования высокого и низкого уровня*
- *Проектирование и архитектура*
- *Двунаправленная трассируемость*
- *Тестирование на основе требований и анализа структурного покрытия*
- *Независимость рассмотрений на более высоких уровнях критичности*
- *Строгое управление конфигурацией, включая сообщения о проблемах*

- *Технические рассмотрения и аудиты гарантии качества по указанным материалам*
- *Строгая гарантия качества для оценки критериев перехода*
- *Квалификация инструментов*
- *Взаимодействие с сертифицирующим органом*

Однако есть и несколько ключевых отличий от DO-178C. Сначала рассмотрим широту области применения DO-278A. Уже из названия видно, что основной фокус – системы реального времени, связанные со связью, навигацией, наблюдением и организацией воздушного движения. Но только ли этим все ограничивается? Нет, поэтому DO-278A неформально называют авиационным стандартом для систем наземного базирования. Какие еще прикладные области могут подпадать под рекомендации DO-278A?

- Наземные пульта/станции управления беспилотными авиационными системами, БАС (Unmanned Aircraft System, UAS), например станции пилота
- Наземное оборудование глобальной системы позиционирования (Global Positioning System, GPS), когда оно относится к контуру управления воздушным судном
- Наземные приемопередатчики, включая функциональность АЗН-В
- Спутниковые системы, поддерживающие связь, наблюдение и навигацию воздушных судов

Если авиационная система расположена на земле или в космосе, обязана ли она соответствовать DO-278A? Не обязательно. Многие наземные системы “связаны” с воздушными судами или авиацией, но не обязаны следовать DO-278A. Примеры: летные тренажеры, управление запасами воздушных судов, планирование полетов, планирование миссий, освещение ВПП и инструменты, оценивающие системы CNS/ATM. Чтобы определить применимость DO-278A, полезно задать два вопроса:

1. “Влияет ли наземная система напрямую на безопасность летящего воздушного судна, не управляя этим воздушным судном напрямую?”
2. “Есть ли у системы выходные данные, которые не были верифицированы другими средствами?”

Если ответ на оба вопроса – “да”, то система, вероятно, относится к сфере DO-278A. Обратите внимание на оговорку в первом вопросе: “не управляя этим воздушным судном напрямую”. В чем смысл? Рассмотрим наземную беспилотную авиационную систему, которая управляет беспилотным воздушным судном, БВС (Unmanned Aerial Vehicle, UAV). В такой системе есть элементы, непосредственно управляющие этим воздушным судном командами реального времени; в этом смысле система ведет себя как пилот, только “пилот” находится на земле. Но команды воздушному судну столь же важны, как если бы

пилот находился на борту, и в таком случае эти элементы могут подпадать под DO-178C. Поэтому границы системы важно обсуждать с сертифицирующими органами уже на раннем этапе планирования проекта.

Рассмотрите следующие утверждения о наземных аэронавигационных системах и DO-278A и определите, истинны они или ложны; ответы и пояснения приведены далее в этой главе:

1. **И / Л:** Все наземные авиационные системы должны применять DO-278A
2. **И / Л:** DO-278A примерно на 50 % похож на DO-178, но между ними существует много важных различий.
3. **И / Л:** Для AL-4 требуются требования низкого уровня, а для AL-5 требуется определенная архитектура
4. **И / Л:** Для AL-3 требуются требования низкого уровня, а для AL-4 требуется определенная архитектура
5. **И / Л:** Требования по независимости есть только у AL-1 и AL-2

Если вы знаете все ответы, поздравьте себя с уровнем знаний DO-278A выше среднего и закрепите эти знания, проверив ответы, которые следуют далее.

Пять планов DO-278A

Как показано выше в материале о трех ключевых процессах DO-278A, сначала идет процесс планирования, а после его завершения – более крупный процесс разработки. На заднем плане непрерывно выполняется самый большой набор интегральных процессов корректности: верификация, управление конфигурацией, гарантия качества и взаимодействие с органом одобрения. Как и в DO-178C, важная задача гарантии качества – убедиться, что планы полны и соответствуют DO-278A, а затем оценивать соответствие инженерных работ этим планам. Кроме того, гарантия качества оценивает критерии перехода между требуемыми мероприятиями, чтобы сроки, входы и выходы каждого мероприятия соответствовали планам. Пять планов DO-278A показаны ниже:

Пять планов DO-278A

1. **План по программным аспектам одобрения** (Plan for Software Aspects of Approval, PSAA).
2. **План гарантии качества ПО** (Software Quality Assurance Plan, SQAP).
3. **План управления конфигурацией ПО** (Software Configuration Management Plan, SCMP).
4. **План разработки ПО** (Software Development Plan, SWDP).
5. **План верификации ПО** (Software Verification Plan, SWVP).

Плюс три стандарта DO-278A: требования, проектирование и кодирование.

Различия между DO-278A и DO-178C

Первое существенное отличие DO-278A – его акцент на системы с готовыми коммерческими компонентами и возможность применять такие компоненты. Тогда как DO-178C возлагает на готовое коммерческое ПО ту же нагрузку, что и на заказное ПО, DO-278A признает, что наземные системы широко используют готовые коммерческие технологии, и потому предусматривает соответствующие механизмы. Что произошло бы, если бы DO-278A не допускал готовое коммерческое ПО? Операционные системы, графика, базы данных и коммуникационные протоколы широко используются в проектах по DO-278A, гораздо шире, чем в бортовой авионике по DO-178C. Кроме того, наземные системы намного богаче функционально, чем бортовые приложения, поэтому объем ПО часто в 10 раз больше. Поскольку готовые коммерческие технологии, как правило, отраслево-нейтральны, они разрабатываются без учета DO-278A; их обратная разработка ради соответствия DO-278A дала бы мало пользы при огромных затратах. Поэтому DO-278A прагматичен: с учетом всего сказанного готовые коммерческие технологии допускаются. Однако применение готовых коммерческих технологий в рамках DO-278A требует:

- заранее определенных стратегий приобретения;
- выявления и анализа верифицируемости;
- верификации интеграции и функциональности;
- строгого управления конфигурацией и контроля;
- соответствия применимым целям DO-278A (с учетом уровня гарантии, на котором используется готовый коммерческий компонент) либо обоснования альтернативных методов определения соответствия.

Обратите внимание: план сертификации в DO-178C назывался “планом сертификации ПО” (Plan for Software Aspects of Certification, PSAC). Такие планы относятся к конкретному проекту, а если проект предназначен для конкретного воздушного судна, то и

соответствующее ПО по DO-178C привязано к проекту и воздушному судну: система считается “сертифицированной” только после сертификации для этого проекта и/или воздушного судна. Системы CNS/ATM устроены иначе: обновления ПО и новая программная функциональность в них появляются более непрерывно, чем на воздушных судах. Кроме того, такие системы в основном состоят из готовой коммерческой аппаратуры. Поэтому системы CNS/ATM “одобряются”, а не сертифицируются; и, в отличие от сертификации аппаратуры воздушных судов по DO-254, аппаратура CNS/ATM проходит испытания на уровне системы, а не как отдельная аппаратура.

Что касается плана по программным аспектам одобрения (Plan for Software Aspects of Approval, PSAA) по DO-278A, по наблюдениям автора, наиболее распространенные ошибки в нем таковы:

- *необоснованный уровень гарантии и отсутствие ссылок на оценки безопасности*
- *отсутствие описания предлагаемой архитектуры системы и ПО*
- *отсутствие описания применяемых инструментов и необходимости/стратегии их квалификации*
- *несоответствие DO-278A, ED-153 (Guidelines for Air Navigation Services (ANS) Software Safety Assurance) и требованиям безопасности*
- *отсутствие одобрения сертифицирующего органа после подачи плана*

Почему DO-278A уделяет так много внимания процессам, связанным с готовыми коммерческими технологиями? Потому что такие технологии настолько распространены, а фактически почти обязательны, в системах CNS/ATM, что сертифицирующие органы хотят видеть связанный с ними верифицируемый процесс качества. Такой процесс начинается с рассмотрения системных требований, которые полностью или частично будут удовлетворяться готовыми коммерческими технологиями. Сначала нужно определить наземную систему с достаточной детализацией, чтобы выполнить анализ “создавать или покупать” по всем требованиям. Требования, отнесенные к варианту “покупать”, становятся требованиями, назначенными готовым коммерческим компонентам. Далее DO-278A требует рассмотреть данные жизненного цикла готовых коммерческих компонентов: будет ли предоставлено достаточно подробностей для оценки? Затем задаются производные требования к функциональности готовых коммерческих компонентов, чтобы явно учесть дополнительные или уточненные возможности, которые должны быть удовлетворены; они также назначаются готовым коммерческим компонентам. Затем необходимо рассмотреть совместимость готовой коммерческой технологии с целевой аппаратно-программной платформой. Для всего этого и для дальнейшего движения по процессу нужно иметь достаточную информацию о готовых коммерческих технологиях.

Еще одна область внимания в DO-278A – готовые коммерческие изделия и безопасность. Поскольку готовые коммерческие изделия применяются шире, нужно отдельно учитывать безопасность применительно к ним. Необходимо описать, как неисправности выявляются, обнаруживаются и снижается их риск. Нужно рассмотреть технические и процедурные границы локализации неисправностей. Через процесс безопасности должны быть выявлены источники ошибок, а затем должны быть предприняты усилия, чтобы доказать их устранение. Помните: принцип “виновен, пока не доказана невиновность” в авиационной сертификации означает, что нужно хранить записи, доказывающие выполнение следующего:

Требования к хранению записей в DO-278A

- Разработан и документирован формальный процесс выполнения анализа безопасности.
- Разработаны требования безопасности, включая производные требования безопасности.
- Использован непрерывный фоновый процесс безопасности, учитывающий влияние всех изменений и анализа результатов тестирования на безопасность.
- Выявлены источники ошибок и доказано их устранение.
- Установлены границы локализации неисправностей, обычно с применением секционирования ПО или системы.

Распространенное заблуждение о DO-278A состоит в том, что крупные и сложные наземные системы, использующие готовое коммерческое ПО и обязанные оставаться экономически эффективными, якобы могут обойтись без затратных процессов низкоуровневого проектирования и анализа кода. **Неверно.** На более высоких уровнях гарантии DO-278A требует требований низкого уровня, проектных решений, рассмотрения кода и анализа структурного покрытия ПО. Почему? Потому что доказано: они необходимы, чтобы помогать оценивать, а значит и повышать, надежность и безопасность.

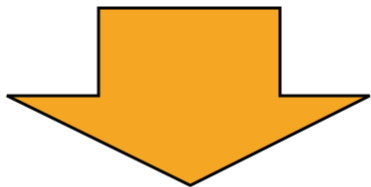
Уровни гарантии (критичности)

В DO-278A есть конкретные цели, зависящие от уровня гарантии (Assurance Level, AL) ПО. Более высокие AL должны удовлетворять большему числу целей DO-278A, чем более низкие уровни. После определения критичности ПО вы обращаетесь к DO-278A, чтобы понять, какие именно цели должно удовлетворять это ПО. Теперь можно переходить к планированию. Здесь DO-278A похож на строительство дома: сначала вы оценили местные условия, чтобы определить нужный тип фундамента, – это ваша “оценка безопасности”. Затем нужен процесс планирования, за ним – процесс разработки.

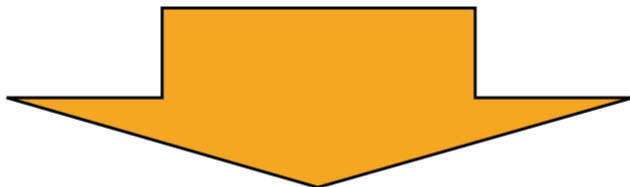
Параллельный процесс корректности идет на протяжении и планирования, и разработки. Поэтому инженерия ПО по DO-278A похожа на строительство дома и следует тому же трехфазному процессному подходу.

Сначала нужно понять уровень гарантии системы и ПО, разрабатываемых по DO-278A. Строгость, применяемая к планированию, разработке и корректности ПО, напрямую связана с его AL, который часто называют “уровнем критичности”. Существует шесть уровней; строгость возрастает от уровня 6 к наиболее строгому уровню 1, как показано ниже. Обратите внимание: “независимость” относится к инженерному процессу верификации, а не к процессу гарантии качества, который независим всегда.

УГ 1 (AL1) <i>Катастроф.</i>	УГ 2 (AL2) <i>Опасное</i>	УГ 3 (AL3) <i>Существенное</i>	УГ 4 (AL4) <i>Суц./ незнач.</i>	УГ 5 (AL5) <i>Незначительное</i>	УГ 6 (AL6) <i>Без влияния</i>
---	--	---	--	---	--



**Независимость
требуется**



Независимость не требуется



Что означает указанная выше “независимость”? Это инженерная независимость, а не независимость гарантии качества, которая требуется всегда. Инженерная независимость означает, что верификация артефакта выполняется людьми и процессами, отличными от автора этого артефакта. Все просто. Только помните, что “верификация” включает рассмотрения, тестирование и анализ.

Теперь сопоставьте это с DO-178C, который обычно известен лучше, чем DO-278A. Пять из шести уровней гарантии DO-278A близко соответствуют пяти уровням DO-178C. Отличие состоит в AL-4, как показано ниже:

DO-278/ED-109	DO-178/ED-12
Уровень гарантии	Уровень ПО
AL1	A
AL2	B
AL3	C
AL4	Нет эквивалента
AL5	D
AL6	E

Как видно из рисунка, в DO-278A есть особый уровень AL-4, расположенный между уровнями C и D стандарта DO-178. Как его рассматривать: как C- или как D+? Ответ: и так, и так. Помните различие между уровнями C и D в DO-178B? У уровня D было 28 целей, тогда как уровень C был намного строже и имел 57 целей. Это означало, что уровень D не интересовался тем, как ПО было спроектировано или реализовано; важно было, чтобы оно выполняло предполагаемую функцию при соблюдении инженерных процессов системного уровня. Поэтому AL-4 стандарта DO-278A находится ровно посередине: он сохраняет умеренную верификацию того, как разрабатывалось ПО. AL-4 требует анализа межкомпонентной связности по данным и по управлению, что относится к проектированию, но НЕ требует ни анализа структурного покрытия ПО, ни тестирования кода на робастность, как это требуется для AL-3.

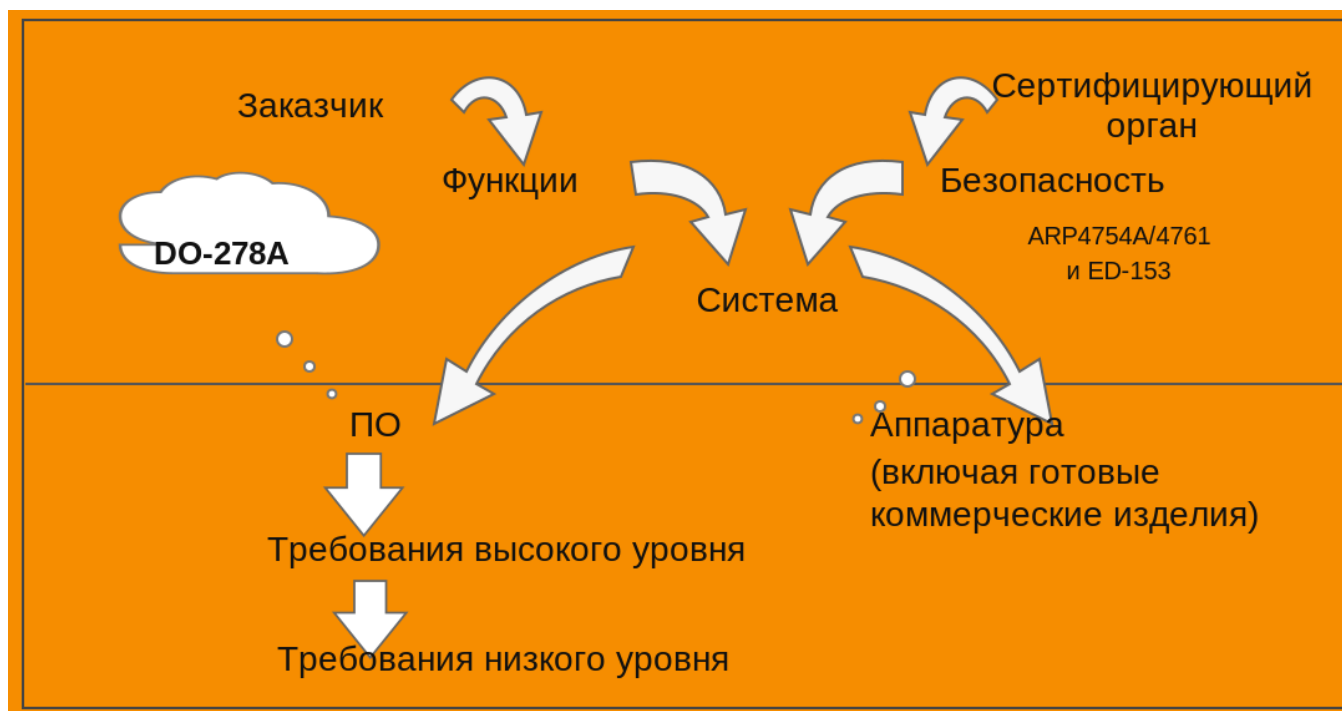
Почему в DO-278 есть такой промежуточный AL-4? Подумайте... большинство систем CNS/ATM поддерживают безопасный полет, но не участвуют в безопасности так непосредственно, как бортовые системы воздушного судна. Например, если дефект системы CNS/ATM случайно сформирует указание пилоту лететь “сквозь” гору, бортовые системы и пилоты с высокой вероятностью обнаружат ошибочное указание и не выполнят его, чтобы избежать столкновения. Поэтому большая часть ПО CNS/ATM попадает в категории AL-3 – AL-5; на AL-1 – AL-2 находится очень мало такого ПО. По опыту автора, 40-50 % ПО CNS/ATM относится к AL-4, где можно широко использовать готовое коммерческое ПО, поскольку AL-4 требует верификации на уровне прикладного программного интерфейса (Application Program Interface, API), но не на уровне внутреннего кода и структур данных. Поэтому ПО AL-4 значительно легче сертифицировать, чем ПО AL-3. На следующем рисунке показаны требуемые цели DO-278A и независимость по уровням гарантии:



Требуемые цели DO-278A и независимость по уровням гарантии

Экосистема требований DO-278A

Как отмечалось в предыдущей главе про авионическое ПО и аппаратуру, основа CNS/ATM также строится на требованиях. Рекомендуемые аэрокосмические практики (Aerospace Recommended Practice, ARP) ARP4754A и ARP4761A были явно разработаны для воздушных судов и бортовых систем, поэтому их применение к CNS/ATM носит скорее вспомогательный характер. В Европе изготовители CNS/ATM и Агентство Европейского союза по авиационной безопасности (European Union Aviation Safety Agency, EASA) все еще в некоторой степени предпочитают применять ED-153, **Руководство по обеспечению безопасности ПО аэронавигационного обслуживания** (Guidelines for Air Navigation Services (ANS) Software Safety Assurance). Однако на момент написания этих строк ARP4754A и ARP4761 постепенно получали все большее признание для применения к CNS/ATM в Европе, тогда как страны вне Европы уже в значительной степени отошли от ED-153. Одна из сложностей ED-153 состоит в том, что его многочисленные руководящие положения считались довольно предписывающими, а значит субъективными и более трудными для оценки. По мнению автора, любому, кто выполняет мероприятия по безопасности CNS/ATM, стоит прочитать ED-153, даже если это формально не требуется: там есть полезная информация о практике оценки безопасности CNS/ATM.



Унаследованные системы CNS/ATM, история эксплуатации и DO-278A

Системы CNS/ATM обычно содержат значительный объем несертифицированного унаследованного ПО и аппаратуры, появившихся до более поздних требований применять DO-278A. Вместо того чтобы начинать заново и перерабатывать эти системы с нуля, повышая вероятность внесения новых ошибок, можно применить концепцию “истории эксплуатации”. Обратите внимание: слово “история” здесь ключевое. Должны существовать доказуемые свидетельства того, что по прежнему ПО или системе велись надежные записи. Меморандум группы сертифицирующих органов по ПО (Certification Authorities Software Team, CAST) N 1 рассматривал атрибуты, применимые к использованию истории эксплуатации; они показаны ниже и взяты из CAST-1. Хотя CAST-1 и многие другие документы CAST были удалены с веб-сайтов FAA и EASA, где они раньше размещались, CAST-1 по-прежнему неформально применяется к системам CNS/ATM. Заметьте, что немногие системы получают полностью “приемлемую” оценку по каждому атрибуту таблицы 3.3-1 CAST-1, поэтому план по программным аспектам одобрения по DO-278A должен содержать подробности по каждому атрибуту. Сама таблица приведена ниже:

Таблица 3.3-1 – Приемлемость атрибутов истории эксплуатации изделия

Атрибут истории эксплуатации изделия	Неприемлемо	Средний уровень	Приемлемо
Продолжительность эксплуатации	Краткая	Средняя	Длительная
Управление изменениями в эксплуатации	Нет	Ограниченное	Полное
Предлагаемое применение по сравнению с применением в эксплуатации	Иное	Сходное	Идентичное
Предлагаемая среда по сравнению с эксплуатационной средой	Иная	Сходная	Идентичная
Число значимых модификаций при эксплуатации	Много	Мало	Нет
Число модификаций ПО при эксплуатации	Много	Мало	Нет
Число модификаций аппаратуры при эксплуатации	Много	Мало	Нет
Возможность обнаружения ошибок	Нет	Частичная	Все ошибки
Возможность сообщения об ошибках	Нет	Частичная	Все ошибки
Число ошибок при эксплуатации	Много	Некоторое число	Нет
Объем/качество доступных и рассмотренных данных истории эксплуатации	Нет/низкое	Некоторый объем/ приемлемое качество	Значительный объем/ высокое качество

Шкала идет слева направо от неприемлемого к приемлемому значению. Легенда исходной таблицы: зачет не допускается; если и допускается, то минимальный; по инженерному суждению зачет не допускается или допускается частично; зачет допускается на основе инженерного суждения; зачет допускается.

Фрагмент CAST-1 по истории эксплуатации изделия

Системы CNS/ATM в большей степени опираются на готовые коммерческие изделия и унаследованное ПО, чем их бортовые аналоги. Бортовое ПО обычно сертифицируется по DO-178C, и организации, применяющие этот стандарт, используют его для оценки соответствия всего исполняемого ПО, включая операционные системы, графические библиотеки, пакеты поддержки плат (Board Support Packages, BSP) и т. д. Однако разработчики систем CNS/ATM и сертифицирующие органы придерживаются несколько иной философии: лучше повторно использовать проверенную технологию, даже если она не разрабатывалась по DO-278A, чем заново создавать полностью новое ПО, которое затем можно сертифицировать по DO-278A. Обоснование простое: наземные системы намного шире используют готовые коммерческие программные и аппаратурные продукты; кроме того, у них есть обширные эксплуатационные кодовые базы, предшествующие DO-278. В результате DO-278A содержит явные положения, допускающие потенциальное использование “альтернативных методов”; затем эти методы становятся альтернативными методами определения соответствия (Alternate Means of Compliance, AMC) для достижения сертифицируемости. В словаре DO-278A определен процесс анализа

пробелов между предлагаемым подходом к обеспечению соответствия и подходом, предписанным DO-278A. После этого такие методы определяются, анализируются на приемлемость и подробно излагаются в плане по программным аспектам одобрения (Plan for Software Aspects of Approval, PSAA) системы CNS/ATM для последующего рассмотрения и одобрения сертифицирующим органом.

Формально установленных или общепринятых альтернативных методов определения соответствия для применения к DO-278A не существует. Однако, по опыту и мнению автора, приемлемые альтернативные методы определения соответствия складываются из следующего “набора инструментов”, включающего шесть категорий таких методов:

Шесть категорий альтернативных методов определения соответствия (Alternate Means of Compliance, AMC)

1. Опыт эксплуатации: один вариант.
2. Дополнительное проектирование ПО: шесть вариантов.
3. Дополнительный процесс ПО: шесть вариантов.
4. Дополнительное рассмотрение или анализ: четыре варианта.
5. Дополнительное тестирование: три варианта.
6. Дополнительная гарантия качества: один вариант.

Опыт эксплуатации (Service experience) почти повсеместно считается первым и самым важным средством альтернативного обеспечения соответствия DO-278A. Но опыт эксплуатации предъявляет высокие требования к доказательствам и должен основываться на многочисленных релевантных критериях: длительности эксплуатации, управлении изменениями, плотности дефектов, истории использования и изменений, а также степени модификаций аппаратуры/ПО, выполненных за этот период или предлагаемых для новой платформы. Эти критерии часто исключают возможность применения опыта эксплуатации.

DO-331: разработка на основе моделей (MBD)

Автор: Вэнс Хилдерман

Рецензенты:

Эрик Диллабер, менеджер по разработке, направление сертификации и стандартов Simulink

Д-р Брюс Дугласс, главный евангелист IBM по направлению интернета вещей (Internet of Things, IoT)

DO-331 и четыре дополнения

DO-331, *Дополнение по разработке и верификации на основе моделей к DO-178C и DO-278A*, представляет собой 125-страничный инструктивный материал, который регулирует применение разработки на основе моделей (Model-Based Development, MBD) в бортовом и наземном авиационном ПО. Поскольку разработка на основе моделей относительно нова для авиационного ПО, перед авторами DO-331 стояла непростая задача: как дать содержательные рекомендации людям, которые в целом не знакомы с этим подходом? В DO-331 этот вопрос решен удачно: практически применимые рекомендации сочетаются с обзорным учебным материалом, который задает общую основу для терминологии и применения разработки на основе моделей.

Спросите любого разработчика ПО (и его руководителя), что является святым Граалем разработки ПО, и, скорее всего, услышите один преобладающий ответ: ПО, автоматически генерируемое из моделей и пригодное к 100% повторному использованию. Справедливости ради, такой ответ обычно дается с улыбкой, а со стороны руководителя инженера, возможно, и с ухмылкой. Но ПО редко бывает полностью автоматически сгенерированным и полностью повторно используемым. В авиации повторное использование ПО применяется широко, и мало кто станет отрицать, что авиационное ПО относится к числу лучших, если не к лучшему, во всем мире критически важного с точки зрения безопасности встраиваемого ПО. Авиационные органы исходят из базовой фактической предпосылки: идеального ПО не существует, и любое ПО в конечном счете может отказать. Этот факт не оспаривается, потому что авиационный ответ на него состоит в обнаружении таких отказов и снижении их последствий. Поэтому разговор о современных практиках разработки ПО неизбежно выводит к разработке на основе моделей как к очередной большой надежде: она должна приблизить нас к этому святому Граалю, то есть к идеальному и идеально повторно используемому ПО...

На первый взгляд DO-331 может показаться интуитивно понятным или почти простым. DO-178C, DO-254 и DO-278A содержат цели, охватывающие весь диапазон инженерии авиационного ПО и аппаратуры, и в целом согласуются с другими стандартами для критически важных с точки зрения безопасности систем. Но важно помнить: DO-331 является дополнением. Как пищевая добавка не заменяет пищу, так и DO-331 дополняет, а не заменяет DO-178C и DO-278A. Основные дополнения семейства DO-XXX таковы:

- **DO-330:** *Квалификация инструмента*
- **DO-331:** *Разработка на основе моделей*
- **DO-332:** *Объектно-ориентированная технология (Object-Oriented Technology, OOT)*
- **DO-333:** *Формальные методы (Formal Methods, FM)*

При использовании любой из этих четырех технологий в авиационной разработке применение соответствующего дополнения обычно обязательно. В этой книге сведения о дополнениях представлены в следующем порядке, поскольку именно такой порядок применения обычно встречается в авиационной разработке:

1. **DO-331:** *Разработка на основе моделей (MBD)*
2. **DO-332:** *Объектно-ориентированная технология (OOT)*
3. **DO-330:** *Квалификация инструмента (TQ)*

Прежде чем углубляться в детали DO-331 по разработке на основе моделей, рассмотрим, почему эти дополнения вообще понадобились, если раньше такие темы разбирались только неформально, через позиционные материалы сертифицирующих органов. Следующий рисунок помогает увидеть, как эволюция программных технологий привела к необходимости этих дополнений:

Почему понадобились дополнения для эволюции программных технологий?

- Прошло почти три десятилетия после выпуска DO-178B.
- Слабые места DO-178B стали заметнее: требования низкого уровня, покрытие, трассировка, связанность.
- Мир неавиационного ПО быстро продвинулся вперед, особенно в следующих областях:
 - разработка на основе моделей (Model-Based Development, MBD);
 - объектно-ориентированная технология (Object-Oriented Technology, OOT);
 - формальные методы (Formal Methods, FM);
 - развитые инструменты.
- Размер и сложность авиационного ПО продолжили расти.
- ПО стало отличительным фактором продукта: его становится больше, и оно становится сложнее.

DO-331 сочетает учебный материал по разработке на основе моделей с рекомендациями по разработке и верификации ПО при использовании моделирования. Хотя документ отчасти интуитивен, его настоящая ценность – в тонкостях, которые позволяют сделать разработку на основе моделей доказуемо детерминированной. С учетом этого рассмотрим следующий простой, на первый взгляд, тест по DO-331 и разработке на основе моделей:

#	Вопрос	Ответ?
1.	<i>Если вы используете моделирование, но не автоматическую генерацию кода (например, пишете код вручную), нужен ли вам все равно стандарт моделирования ПО и анализ покрытия модели?</i>	(Да, Нет или Может быть)
2.	<i>При использовании моделей для верификации должны ли ожидаемые результаты быть определены до выполнения теста?</i>	(Да, Нет или Может быть)
3.	<i>Можно ли создать проектную модель (Design Model) непосредственно на основе системных требований, не выполнив предварительно декомпозицию этих требований в требования к ПО высокого уровня?</i>	(Да или Нет)
4.	<i>Должны ли системные требования быть сначала декомпозированы в требования к ПО высокого уровня до создания проектной модели?</i>	(Да или Нет)
5.	<i>Если системные требования используются как источник требований для проектной модели, рассматриваются ли такие системные требования как требования высокого уровня и содержит ли проектная модель требования низкого уровня?</i>	(Да, Нет или Может быть)

Ответы на эти вопросы даны в этой главе.

Зачем нужно моделирование?

Моделирование ПО почти так же старо, как и сама разработка ПО: модели активно помогали инженерам еще во время ранних лунных запусков Национального управления по аэронавтике и исследованию космического пространства США (NASA). Сегодня моделирование означает общий вид деятельности, при котором структура и поведение объектов, а также взаимодействия между объектами задаются на более высоком уровне абстракции, чем разработка логики. В обычном употреблении термин “модель” подразумевает, что сначала создается модель структуры и поведения, а затем эта модель используется для генерации фактической программной или аппаратной логики. Напомним: в мире авионики термин “аппаратура” включает логику, физически реализованную в интегральной схеме. Раньше такую логику часто называли термином *firmware*, но по смыслу здесь речь не о программном коде в энергонезависимой памяти, а об аппаратной логике; теперь в DO-254 она относится к сложной электронной аппаратуре. Однако моделирование может не только предшествовать разработке логики, но и следовать за ней. Ко многим унаследованным кодовым базам моделирование применяют постфактум, чтобы облегчить понимание, верификацию, улучшение или повторное

использование этих баз кода. Это особенно характерно для авионики, где большинство систем хотя бы частично повторно использует предшествующие похожие системы. Такой процесс обычно называется обратной разработкой.

Почему моделирование становится все важнее для авионики, что подтверждается выпуском DO-331 в декабре 2011 года? Причин много, включая перечисленные ниже:

Преимущества разработки на основе моделей

1. **Улучшенная работа со сложными системами.** Моделирование улучшает понимание и управление все более сложными системами.
2. **Повышенная возможность повторного использования.** Моделирование способствует повторному использованию за счет абстрагирования функциональности.
3. **Возможность автоматической генерации кода.** Моделирование позволяет автоматически генерировать код непосредственно из модели.
4. **Улучшенная ясность логики.** Моделирование позволяет применять графические языки, такие как Simulink или язык моделирования систем (Systems Modeling Language, SysML).
5. **Общий язык означает меньше допущений.** Системные инженеры и разработчики используют общий язык моделирования.
6. **Улучшенная трассируемость.** Инструменты моделирования могут автоматизировать трассировку от требований к проектным решениям и коду.
7. **Более ранняя верификация.** Симуляция модели позволяет раньше выполнять валидацию требований и верификацию проектных решений.

Как видно из рисунка “Преимущества моделирования”, моделирование сложных систем и ПО дает много преимуществ. В других областях ПО, например в телекоммуникациях, моделирование используется уже много лет; авионика постепенно догоняет. Следует отметить, что руководящие материалы по сертификации авиационного ПО в целом не особенно интересуются стоимостью и сроками. Никто не хочет и не ожидает, что разработчики авионики обанкротятся, но формально им это не запрещено. Что же тогда мешает всем дружно внедрить моделирование в организациях авиационной разработки? Ниже перечислены причины, которые часто приводят разработчики авионики, избегая частичного или полного использования моделирования:

1. *Неопределенность в толковании DO-331 или опасение, что будет применено чрезмерно консервативное толкование;*
2. *Стоимость лицензий на инструменты моделирования;*
3. *Сложность освоения инструментов моделирования;*

4. *Представление о сложности или необходимости квалифицировать инструмент моделирования по DO-330;*
5. *Культура традиционных системных инженеров, которые предпочитают работать только с английским текстом и избегают использования инструментов моделирования или ввода данных в такие инструменты;*
6. *Страх делать что-либо новым способом;*
7. *Недоверие к качеству исходного кода, сгенерированного из моделей;*
8. *Опасение, что разработчик не контролирует конкретные операторы исходного кода, которые генерируются из моделей.*

Обоснованы ли эти причины отказа от моделирования? В той мере, в какой для некоторых людей восприятие равно реальности, ответ будет: может быть. Но при более информированном понимании это восприятие можно изменить. Во-первых, DO-331 задает основу для применения моделирования в рамках предписанного набора стандартов моделирования и мероприятий по верификации моделей. Как традиционные требования к ПО и проектные решения ПО должны проходить верификацию для подтверждения соответствия стандартам, заданным пользователем, так и авиационные модели должны проходить такую верификацию. Во-вторых, стоимость лицензий на инструменты моделирования на самом деле не так велика, если учитывать экономию за счет роста инженерной производительности. Некоторые инструменты моделирования, например IBM Rhapsody и MagicDraw, вполне доступны по цене и при этом дают необходимые ключевые возможности. В-третьих, как и с лицензионными затратами, сложность освоения инструментов моделирования обычно преодолевается за недели или, самое большее, за несколько месяцев. На фоне многолетней разработки большинства авиационных систем это не так много. В-четвертых, квалифицировать инструмент моделирования не обязательно. Преимущества квалификации существенны, например можно исключить ручные коллегиальные рассмотрения кода, использовать автоматическую генерацию кода и верификацию на уровне модели. Но все преимущества моделирования, перечисленные выше, сохраняются независимо от того, квалифицирован инструмент моделирования или нет. В-пятых, инженеры способны осваивать новые методы, и большинство действительно приветствует такие знания, в том числе понимая их пользу для резюме. Поэтому кажущиеся недостатки моделирования вполне можно развеять. Более того, Simulink, IBM Rhapsody и инструменты Ansys SCADE существуют уже около 20 лет и достаточно зрелы как по качеству генерируемого кода, так и по возможностям управления этой генерацией и ее настройке. Тем не менее моделирование по DO-331 требует внимания к ключевым атрибутам, кратко показанным на следующем рисунке и далее рассмотренным в этой главе:

Атрибуты разработки на основе моделей, соответствующей DO-331

1. Стандарты моделей, включая спецификационные и проектные модели.
2. Требования к моделям.
3. Единицы конфигурации моделей.
4. Библиотеки элементов моделей.
5. Интерфейсы моделей и системы.
6. Данные указателя конфигурации моделей.
7. Среда моделирования и руководство пользователя.

Особенно важно при моделировании соблюдать стандарт моделирования и затем оценивать такое соблюдение. В отличие от стандартов кодирования ПО, например правил MISRA C, упомянутых в предыдущей главе о DO-178C, DO-331 не ссылается на готовые явные стандарты моделирования. Тем не менее стандарт моделирования, соответствующий DO-331, должен задавать оцениваемые критерии по следующим ключевым темам:

Ключевые аспекты разработки на основе моделей (Model-Based Development, MBD), охватываемые стандартами моделирования

1. Методы моделирования.
2. Язык моделирования.
3. Инструменты моделирования.
4. Руководства по моделированию.
5. Ограничения моделирования.
6. Идентификация различных элементов модели.

Терминология моделирования

У моделирования есть собственный профессиональный язык; основные термины моделирования кратко приведены ниже.

Терминология моделирования DO-331, часть 1

- **Генерация кода.** Генерация кода из проектной модели на заданном языке исходного кода ПО, таком как C, C++ или Ada.
- **Проектная модель (Design Model).** Модель, определяющая проект ПО: требования низкого уровня, архитектуру ПО, алгоритмы, внутренние структуры данных компонентов, поток данных и/или поток управления. Модель применяется для генерации исходного кода ПО.
- **Модель.** Абстрактное представление заданного набора аспектов системы, используемое для анализа, верификации, симуляции, генерации кода или их комбинации. Представление должно быть однозначным независимо от уровня абстракции. Примечание: заданный набор аспектов может содержать все аспекты системы или только ее подмножество.
- **Разработка и верификация на основе моделей.** Технология, при которой модели представляют требования к ПО и/или описания проекта ПО для поддержки процессов разработки и верификации ПО.
- **Тест на основе модели.** Создание тестовых примеров в языке и инструменте моделирования для верификации модели и/или сгенерированного исходного кода.

Терминология моделирования DO-331, часть 2

- **Проверка модели.** Применение правил корректного построения для обеспечения синтаксической корректности модели, например удобочитаемости состояний и надлежащего использования элементов языка моделирования.
- **Анализ покрытия модели.** Анализ, определяющий, какие требования, выраженные проектной моделью, не были задействованы верификацией на основе требований, из которых эта проектная модель была разработана. Цель состоит в обнаружении непредусмотренной функциональности в проектной модели.
- **Элемент модели.** Единица, из которой строится модель.
- **Библиотека элементов модели.** Набор элементов модели, используемый как основа для построения модели. Модель может разрабатываться с использованием библиотек элементов модели или без них.
- **Симуляция модели (Model Simulation).** Действие по выполнению поведения модели с использованием симулятора модели.

Терминология моделирования DO-331, часть 3

- **Симулятор модели.** Устройство, компьютерная программа или система, обеспечивающие выполнение модели для демонстрации ее поведения в поддержку верификации и/или валидации. Симулятор модели может выполнять или не выполнять код, репрезентативный для целевого кода.
- **Методика моделирования.** Сочетание языка моделирования и определенного порядка использования этого языка. Определяется уровнем абстракции информации, представляемой моделью, и выбранными инструментами моделирования.
- **Формирование отчета.** Создание документов по модели для конкретной цели, часто на основе шаблонов, для формирования документов в стандартизированном формате.
- **Обратная разработка.** Создание модели по исходному коду.
- **Спецификационная модель (Specification Model).** Модель, представляющая требования высокого уровня и дающая абстрактное представление функциональности, производительности, интерфейса ПО или характеристик его безопасности. Исключает внутренние структуры данных, внутренний поток данных или внутренний поток управления.

Терминология моделирования DO-331, часть 4

- **Символика.** Графический внешний вид элементов моделирования. Некоторые среды моделирования позволяют задавать пользовательскую символику.
- **Язык моделирования систем (Systems Modeling Language, SysML).** Специализированный вариант унифицированного языка моделирования для применения при моделировании систем.
- **Унифицированный язык моделирования (Unified Modeling Language, UML).** Наиболее распространенный язык моделирования ПО. Унифицированный язык моделирования содержит языковые элементы для задания структуры, поведения, функциональности и связей ПО.

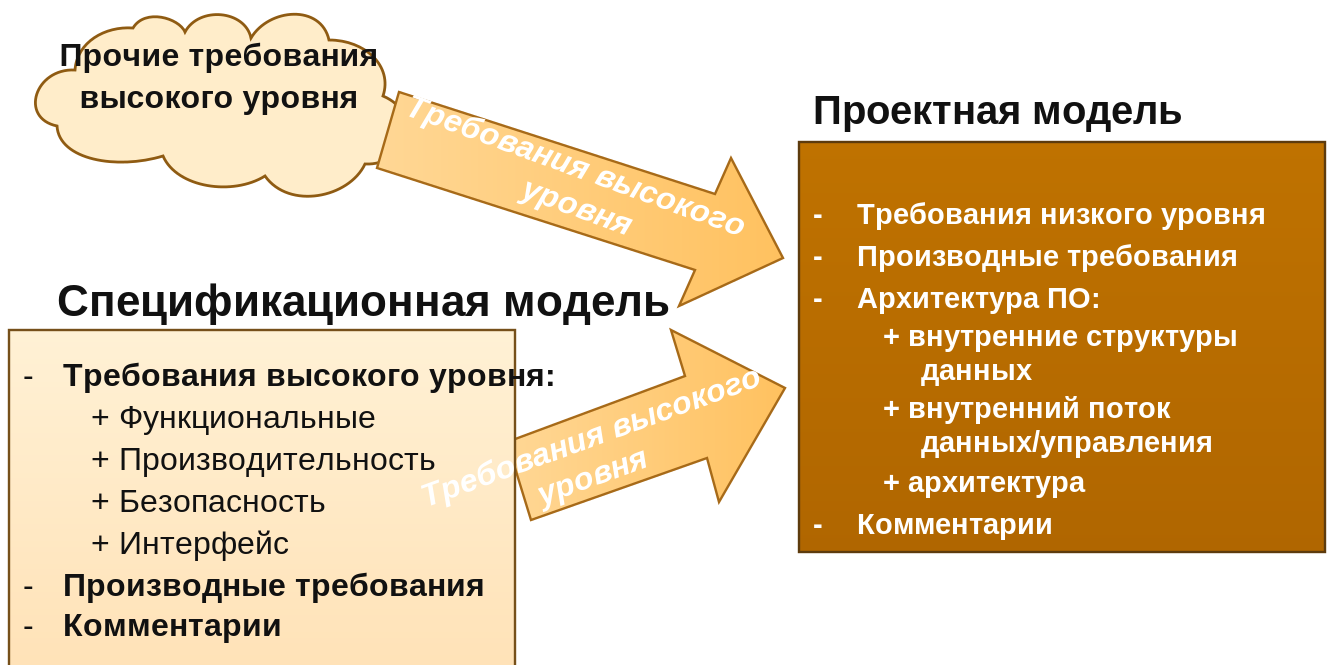
Спецификационная и проектная модели

Важно помнить базовый принцип DO-178C: пошаговое уточнение. Системные требования должны предшествовать требованиям высокого уровня (High-Level Requirements, HLR), а требования высокого уровня – требованиям низкого уровня (Low-Level Requirements, LLR). Следует избегать соблазна разрабатывать код непосредственно из требований высокого уровня. Моделирование позволяет выражать требования высокого уровня и/или требования низкого уровня прямо в модели. Ключевой аспект моделирования по DO-331 – различие между спецификационной моделью (Specification Model) и проектной моделью (Design Model), как показано на следующем рисунке:

Спецификационная модель в DO-331 в сравнении с проектной моделью

- **Спецификационная модель.** Обычно выражает требования высокого уровня (High-Level Requirements, HLR).
- **Проектная модель.** Обычно выражает требования низкого уровня (Low-Level Requirements, LLR) и может использоваться для получения кода.

Спецификационная и проектная модели различаются так же, как требования высокого уровня отличаются от требований низкого уровня: это пошаговое уточнение. Как требования высокого и низкого уровня могут находиться в одной спецификации требований, так и спецификационная и проектная модели могут находиться в одной общей модели. Но если существуют обе модели, спецификационная модель разрабатывается раньше проектной модели. Поскольку эти модели решают разные задачи, для каждой из них должен применяться свой стандарт моделирования.

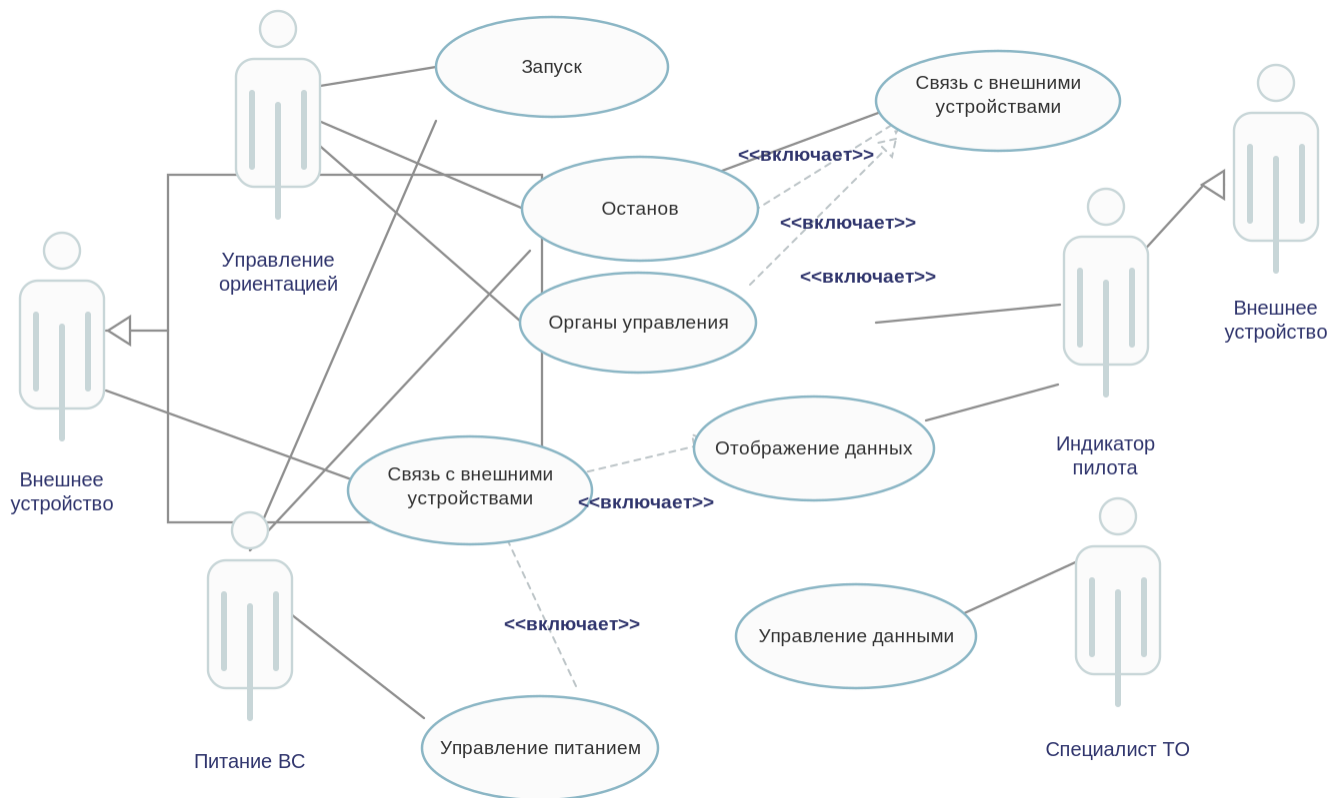


В унифицированном языке моделирования (Unified Modeling Language, UML) спецификационная модель обычно использует варианты использования для группировки требований, а затем применяет различные механизмы UML: конечные автоматы, моделирование сценариев и моделирование деятельности. Это уточняет требования и устраняет неоднозначность. Проектная модель определяет требования низкого уровня, необходимые для удовлетворения уточненных требований и спецификационной модели. Связь “trace” поддерживает трассируемость между требованиями, спецификационной моделью и проектной моделью.

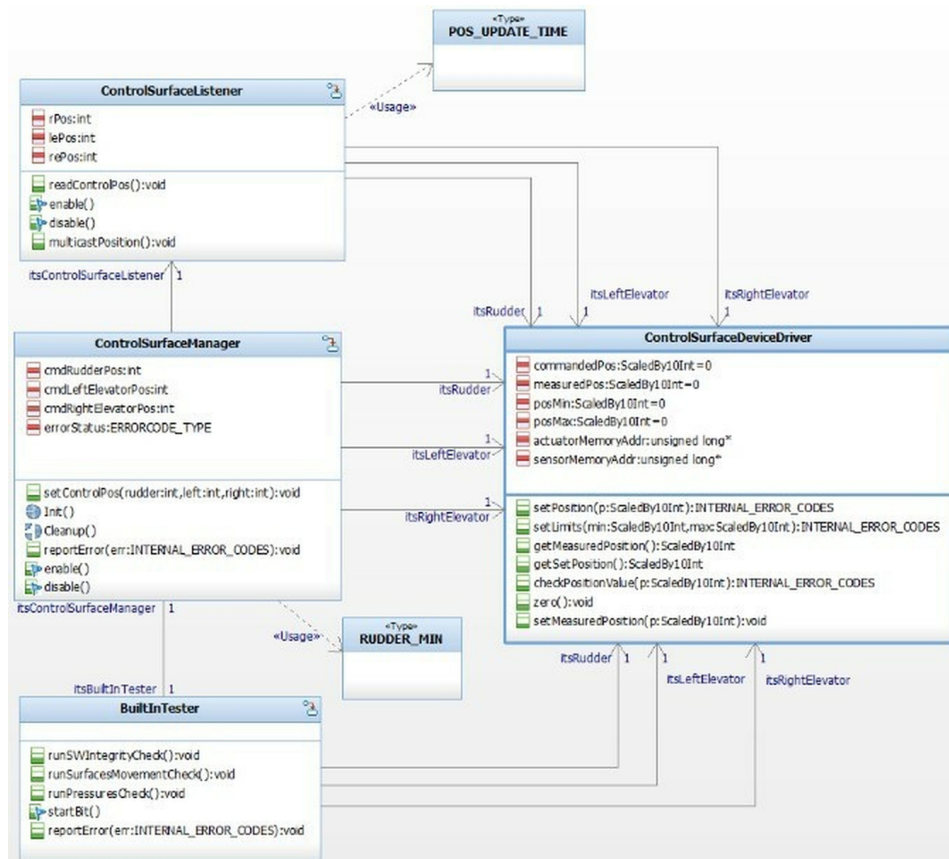
Процесс разработки модели не задан ни в DO-178C, ни в DO-331, хотя эти стандарты определяют цели, которым такие процессы должны соответствовать, и свидетельства, которые они должны формировать. Один из распространенных процессов – процесс

Harmony для встроенного ПО (Harmony Process for Embedded Software). См. *Real-Time Agility* или *Real-Time UML Workshop for Embedded Systems*, 2-е издание; обе книги написаны Брюсом Пауэлом Дуглассом.

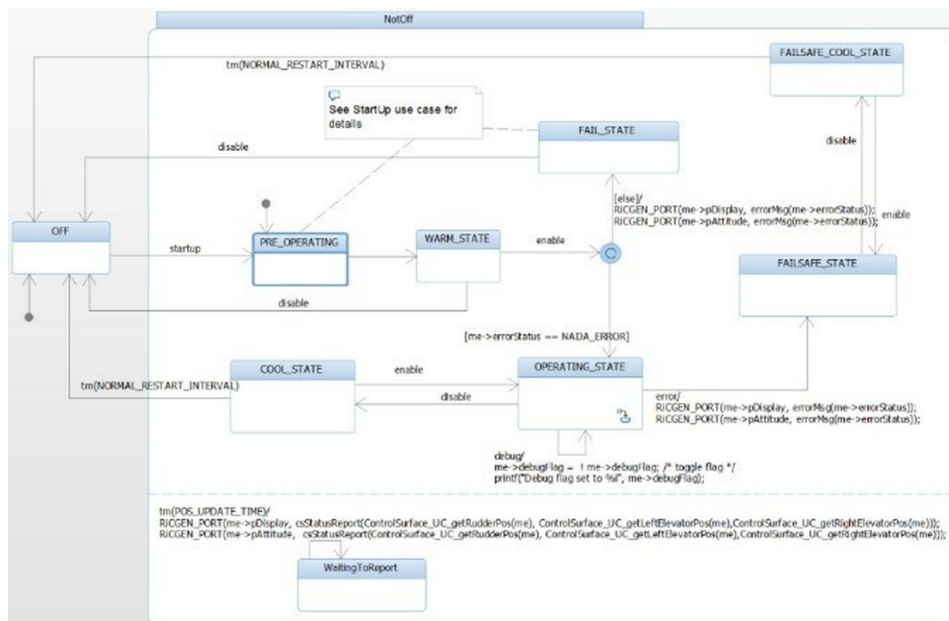
Пример диаграммы вариантов использования (повторно использовано с разрешения д-ра Брюса Дугласса, IBM):



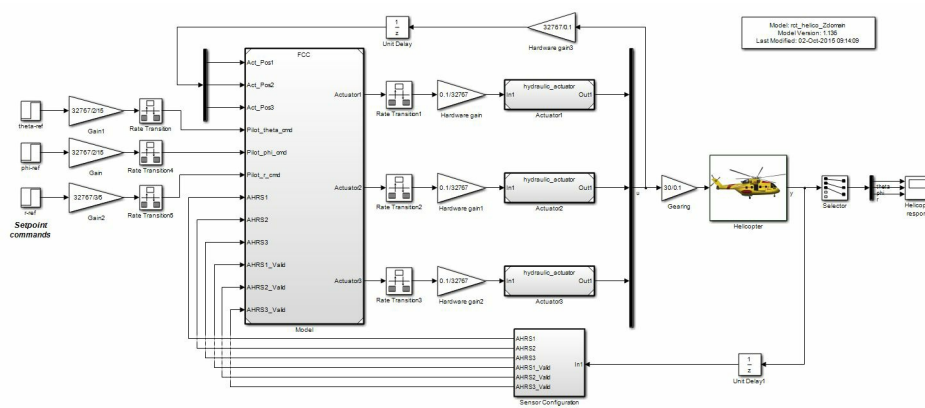
Пример диаграммы классов (повторно использовано с разрешения д-ра Брюса Дугласса, IBM):



Пример диаграммы состояний (повторно использовано с разрешения д-ра Брюса Дугласса, IBM):



Пример модели Simulink (повторно использовано с разрешения г-на Эрика Диллабера, менеджера по сертификации Simulink):



Выбор стратегии моделирования

Как искусство, музыка и высокая кухня, моделирование означает разные вещи для разных людей, и реализовывать его можно очень по-разному. Чтобы сузить множество возможных способов применения моделирования, DO-331 описывает пять вариантов моделирования и рекомендует выбрать один из них. На следующем рисунке показаны пять рекомендуемых вариантов моделирования, от MB1 до MB5.

Процесс, формирующий данные жизненного цикла	MB1	MB2	MB3	MB4	MB5
Процессы системных требований и проектирования системы	Системные требования	Системные требования	Системные требования	Системные требования / требования высокого уровня (HLR)	Системные требования / требования высокого уровня (HLR) Проектная модель (требования низкого уровня, LLR)
Процессы требований к ПО и проектирования ПО	Требования высокого уровня (HLR) Проектная модель (требования низкого уровня, LLR)	Спецификационная модель (требования высокого уровня, HLR) Проектная модель (требования низкого уровня, LLR)	Спецификационная модель (требования высокого уровня, HLR) Требования низкого уровня (LLR)	Проектная модель (требования низкого уровня, LLR)	
Процесс кодирования ПО	Исходный код	Исходный код	Исходный код	Исходный код	Исходный код

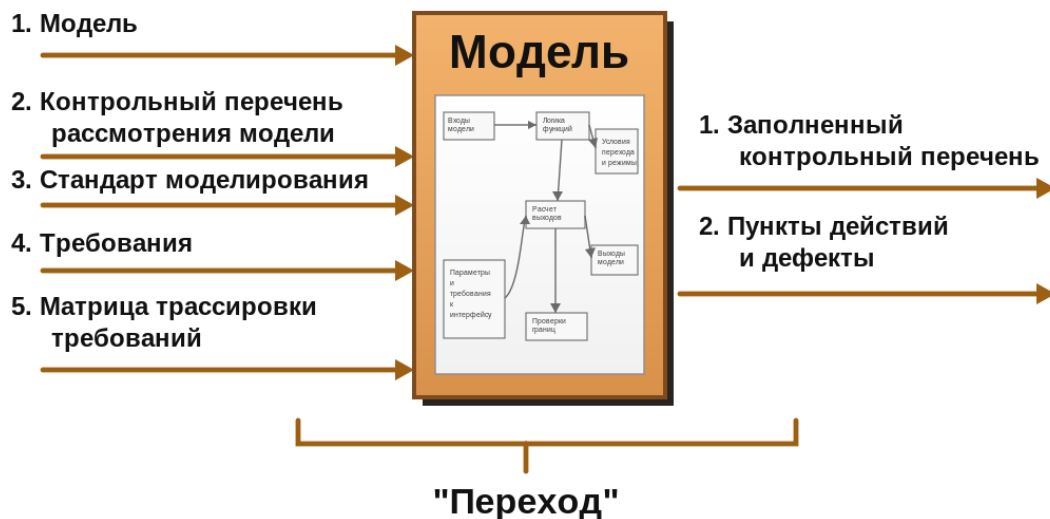
Как и в большинстве жизненных решений, здесь есть выбор, и один вариант подходит не всем. Но если учитывать, откуда вы пришли, где находитесь и куда направляетесь, предпочтительный вариант определить гораздо проще. Преимущества и недостатки каждого варианта моделирования кратко приведены ниже:

Преимущества и недостатки вариантов моделирования MB1-MB5

- **MB1: традиционная разработка с проектной моделью.**
 - **Преимущества:** вводит моделирование в разработку ПО; обеспечивает генерацию автокода.
 - **Недостатки:** разделяет системную инженерию и инженерию ПО; требования высокого уровня (High-Level Requirements, HLR) не зависят от модели.
- **MB2: традиционная разработка со спецификационной и проектной моделями.**
 - **Преимущества:** хорошо использует спецификационные и проектные модели; поддерживает генерацию автокода.
 - **Недостатки:** сохраняется разделение системной инженерии и инженерии ПО.
- **MB3: традиционная разработка со спецификационной моделью.**
 - **Преимущества:** вводит моделирование для требований высокого уровня (High-Level Requirements, HLR).
 - **Недостатки:** нет генерации автокода; возможно снижение мотивации сопровождать модель.
- **MB4: требования высокого уровня объединены с системными требованиями; инженеры ПО разрабатывают проектную модель.**
 - **Преимущества:** усиливает вклад системной инженерии в требования к ПО; поддерживает генерацию автокода.
 - **Недостатки:** нет спецификационной модели; сложные системы могут пропустить пошаговое уточнение.
- **MB5: требования высокого уровня объединены с системными требованиями; системные инженеры разрабатывают проектную модель.**
 - **Преимущества:** системно-центричная разработка и контроль требований высокого уровня и модели; минимум неоднозначности.
 - **Недостатки:** нет спецификационной модели; возможен слишком большой шаг от требований к коду.

Входные данные для моделирования: верификация модели

В некоторых областях разработки ПО проектировщики начинают с чистого листа или концепции, а затем итеративно развивают соответствующую модель ПО. Но в авиации действует парадигма “виновен, пока не доказана невиновность”: инженерной работе не доверяют, пока она не квалифицирована или не верифицирована. Поэтому рассмотрим следующий вопрос: модель нужно верифицировать; можно ли верифицировать модель одним лишь рассмотрением? Конечно, нет. Верификация модели требует формального рассмотрения нескольких входных данных, как показано ниже:



Какие из пяти входных данных действительно нужны для обязательного рассмотрения модели? Все пять. Каждый из них должен находиться под управлением конфигурацией, то есть иметь уникальный идентификатор и быть доступным для точного извлечения в любой момент в будущем. Это нужно, чтобы можно было определить точное содержание, использованное во время соответствующего рассмотрения модели. Обратите внимание: распространенная сложность при моделировании – верификация требований. Повторим: у моделей должны быть требования, то есть функциональность ПО, которые используются при рассмотрении модели для оценки ее корректности и полноты.

Верификация посредством тестирования

Помимо рассмотрения, которое оценивает соответствие модели применимому стандарту моделирования и системным требованиям / требованиям к ПО, сами модели также могут верифицироваться посредством тестирования. Профиль UML для тестирования (UML Profile for Testing) определяет стандартный способ спецификации тестовых примеров, построения архитектуры тестов, их выполнения и анализа. Это можно делать вручную, создавая элементы тестирования и используя симуляцию или исполнение модели, чтобы убедиться, что результаты соответствуют ожиданиям. Такие инструменты, как надстройка Test Conductor для IBM Rhapsody, автоматизируют часть этих действий и также могут применяться.

Верификация посредством формальных методов (DO-333)

Формальные методы – еще одно ключевое средство верификации моделей, особенно подмножеств системных моделей. Некоторые инженеры пытаются верифицировать формальными методами целые системные модели, и DO-333, судя по всему, это допускает. DO-333 является дополнением, обычно применяемым совместно с DO-178C и DO-278A. Однако лучшая практика состоит в другом: сосредоточить верификацию модели формальными методами на конкретном алгоритме или наборе связанных алгоритмов внутри модели, чтобы максимально повысить доказуемость. Дополнительную информацию см. в разделе о формальных методах.

Симуляция модели

Еще одно преимущество моделирования – симуляция: симулятор модели может исполнять модель на более раннем этапе жизненного цикла разработки. Такая симуляция может помогать верификации следующим образом:

- *Соответствие системным требованиям (для спецификационных моделей), соответствие требованиям высокого уровня (для проектных моделей), точность и согласованность*
- *Верифицируемость*
- *Аспекты алгоритма*

Однако симуляция модели не предназначена для верификации следующего:

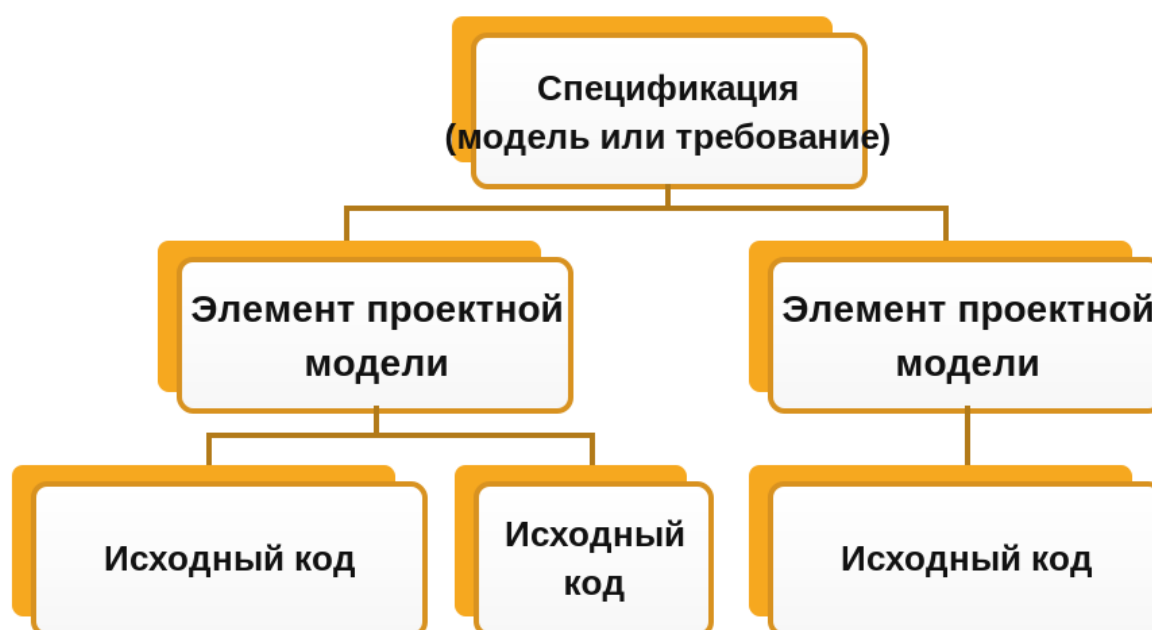
- *Совместимость с целевым вычислителем*
- *Соответствие стандартам*
- *Трассируемость*
- *Целостность обособления*

Верификация модели и кода

Профиль UML для тестирования (www.omg.org) предоставляет стандартный подход к спецификации, выполнению и анализу тестовых примеров на языке UML. Это означает, что преимущества моделирования можно получать не только от спецификационных и проектных моделей, но и от способов, которыми эти модели верифицируются. Надстройка Test Conductor для IBM Rhapsody реализует этот стандарт и поставляется с комплектом квалификации по DO-178. Этот инструмент автоматизирует генерацию, выполнение и анализ тестов, может предоставлять подробную статистику покрытия модели (см. следующий раздел), а при сопряжении с инструментами верификации кода – также покрытие на уровне кода.

Анализ покрытия модели и трассируемость

Модели разрабатывают инженеры, а инженеры могут забыть добавить необходимые детали модели или удалить детали, которые уже не нужны. Поэтому необходимо выполнять анализ покрытия модели. Такой анализ позволяет определить, какие элементы модели могли быть верифицированы не полностью, а также обнаружить непредусмотренную функциональность или неверифицированные элементы внутри модели. Анализ покрытия модели может выполняться с помощью симуляции или формальных методов; анализ структурного покрытия ПО, однако, выполняется фактическим тестированием. На схеме ниже показаны элементы, которые учитываются при анализе покрытия модели через трассируемость: спецификация, то есть модель или требование, связана с элементами проектной модели, а элементы проектной модели связаны с реализующим их исходным кодом.



Элементы модели, учитываемые при обязательном анализе покрытия модели

Трассируемость модели также необходима, чтобы доказать: каждый элемент модели присутствует по определенной причине. Каждая часть модели должна трассироваться на требование или требования, которые она реализует, и на исходный код, который реализует ее.

Трассируемость модели, разумеется, можно вести вручную, но инструменты моделирования все чаще имеют встроенные возможности для поддержки и обеспечения трассируемости.

Заключение

Моделирование – мощная возможность, которая все шире применяется в авионике. DO-331 дает основу для понимания моделирования, использования его возможностей и доказательства корректности модели.

Ответы на вопросы, поставленные ранее в этой главе, приведены ниже.

#	Вопрос	Ответ?
1.	<i>Если вы используете моделирование, но не автоматическую генерацию кода (например, пишете код вручную), нужен ли вам все равно стандарт моделирования ПО и анализ покрытия модели?</i>	Да
2.	<i>При использовании моделей для верификации должны ли ожидаемые результаты быть определены до выполнения теста?</i>	Да
3.	<i>Можно ли создать проектную модель непосредственно на основе системных требований, не выполнив предварительно декомпозицию этих требований в требования к ПО высокого уровня?</i>	Да
4.	<i>Должны ли системные требования быть сначала декомпозированы в требования к ПО высокого уровня до создания проектной модели?</i>	Нет
5.	<i>Если системные требования используются как источник требований для проектной модели, рассматриваются ли такие системные требования как требования высокого уровня, а проектная модель содержит требования низкого уровня?</i>	Да

DO-332. Объектно-ориентированная технология (Object-Oriented Technology, OOT)

Автор: Вэнс Хилдерман

Рецензент:

г-н Брайан Аллен, ведущий инженер по ПО, Sikorsky, компания Lockheed Martin

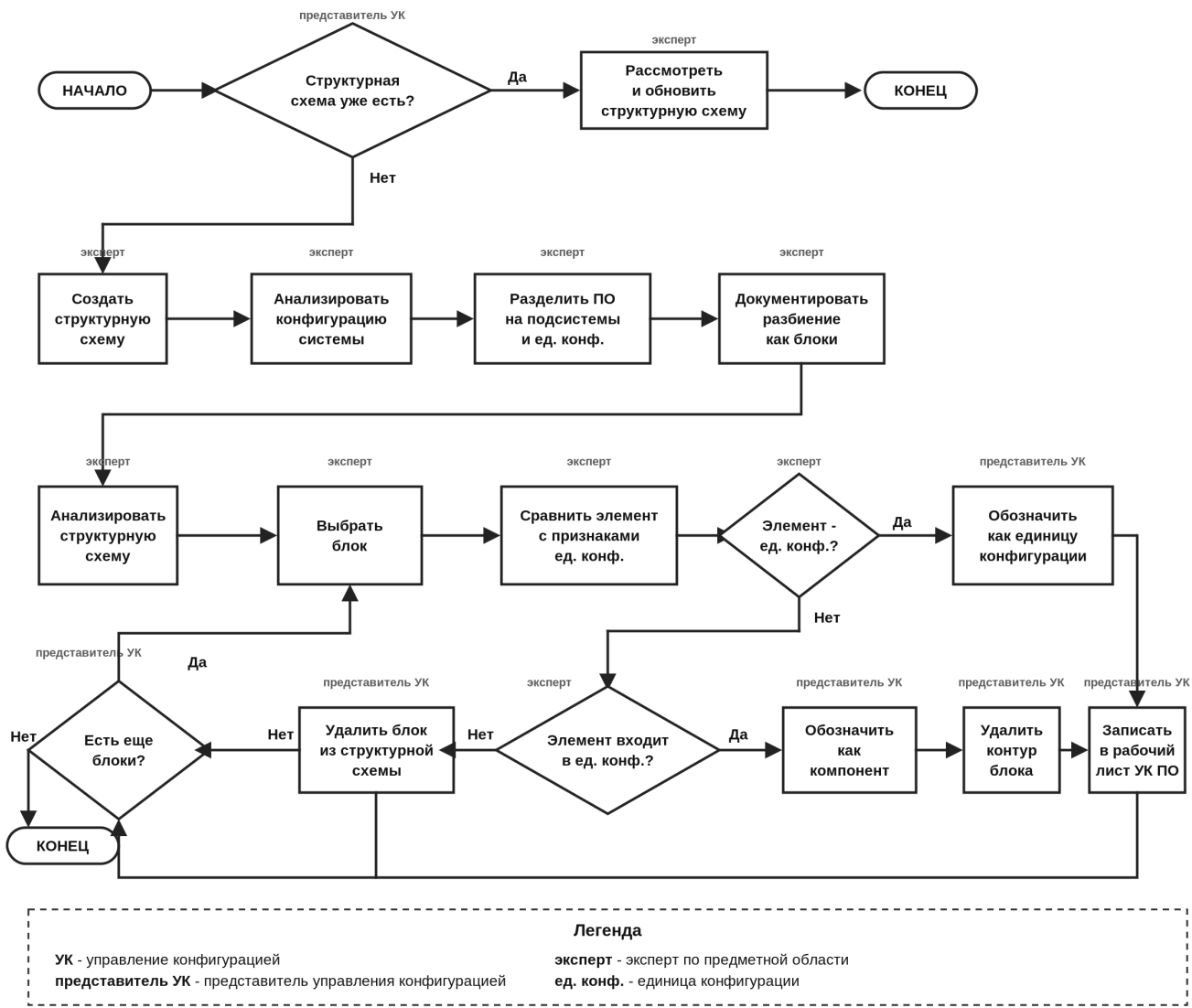
Введение и история DO-332 и объектно-ориентированной технологии (OOT)

DO-332, *Дополнение по объектно-ориентированной технологии и связанным методам к DO-178C и DO-278A (Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A)*, представляет собой 150-страничное руководство по применению объектно-ориентированной технологии в бортовом и наземном авиационном ПО. Но поскольку настоящая объектно-ориентированная технология для авиационного ПО появилась сравнительно недавно, хотя Ada 95 существует с 1995 года, перед авторами DO-332 стояла непростая задача: как дать содержательные рекомендации людям, которые в целом не знакомы с объектно-ориентированным ПО? В DO-332 это сделано удачно: практические рекомендации объединены с введением в объектно-ориентированную технологию, которое создает общую основу для терминологии, применения и сертифицируемости.

Существует много подходов к разработке ПО. Изданы сотни книг, и многие из них, похоже, проповедуют собственную “методологию”. Но как многочисленные цвета павлина получаются из базовых красного, зеленого и синего, так и у разработки ПО, если смотреть предельно упрощенно, есть два “основных цвета”: функциональная структурированная реализация и объектно-ориентированная реализация. В отличие от павлина, в ПО эти “цвета” плохо смешиваются: многие элементы проекта ПО считаются либо функциональными, либо объектно-ориентированными, но не тем и другим одновременно. Традиционное функциональное ПО реализуют, последовательно рассматривая и структурируя каждую цепочку действий компьютера. Объектно-ориентированное ПО, напротив, проектируют, сначала определяя программные объекты и действия над ними, а затем объединяя эти объекты и действия в осмысленные группы и события. Да, в функциональном проекте могут быть объекты. Да, объектно-ориентированный проект использует некоторые последовательные структурные

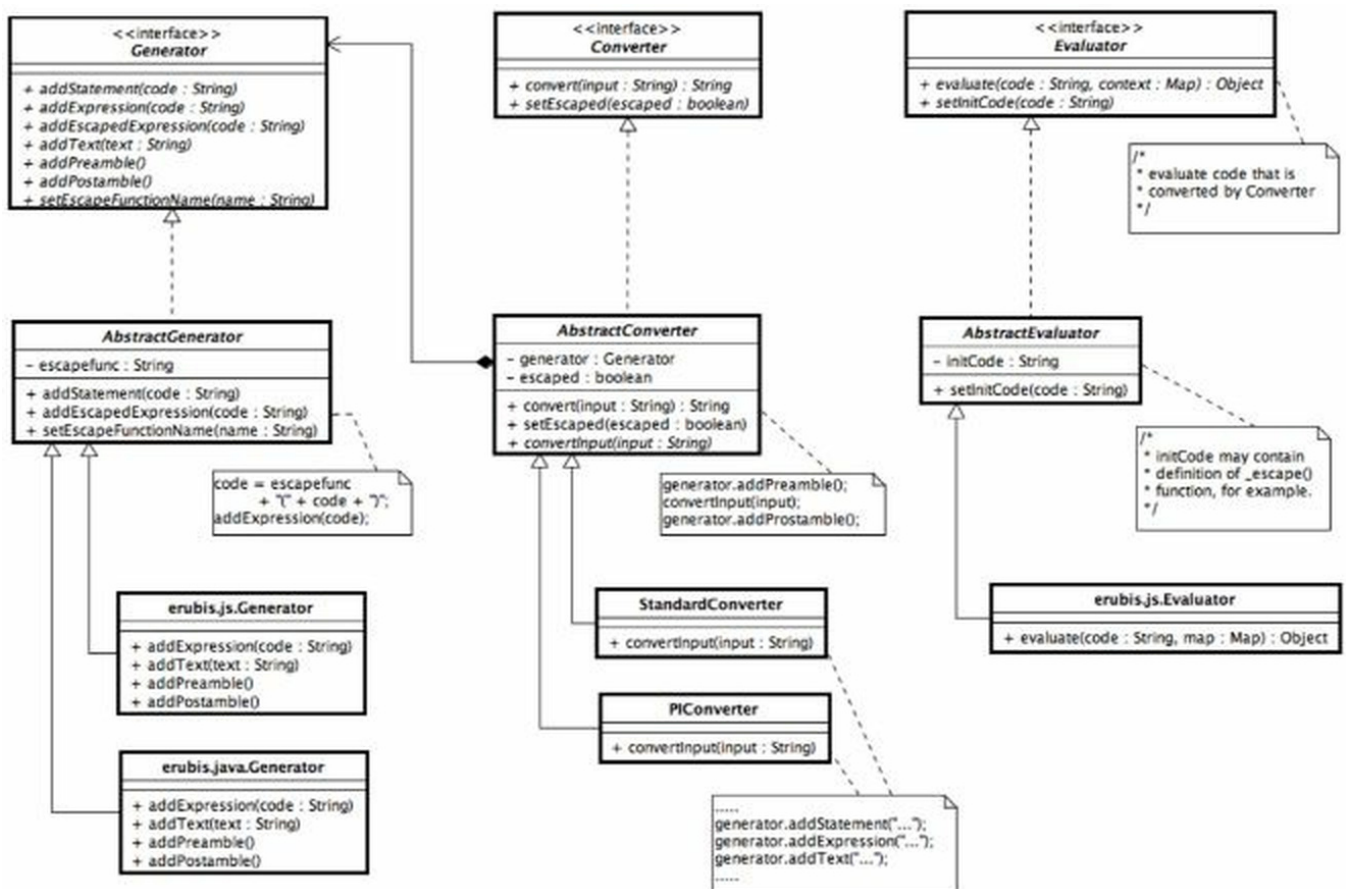
поведения. Но несколько капель масла могут плавать на воде, не становясь с ней единым целым; так же функциональное и объектно-ориентированное проектирование остаются двумя разными подходами, которые в чистом виде трудно интегрировать друг с другом.

До публикации DO-332 у разработчиков критичного по безопасности ПО было мало правил применения объектно-ориентированной технологии. Стандарты программирования, такие как стандарты Ассоциации по надежности ПО в автомобильной промышленности для C/C++ (Motor Industry Software Reliability Association, MISRA), уже существовали и были полезны. Их использовали, и их по-прежнему следует применять вместе с коммерческим инструментом статического анализа, чтобы очищать код C/C++ и повышать его качество. Однако четких указаний по проектированию и верификации критичного по безопасности объектно-ориентированного ПО не хватало; DO-332 пытается закрыть этот пробел. В функциональном проектировании ПО поток управления заранее задается разработчиком, поэтому последовательность решений, то есть поток управления (control flow) в DO-178C, рассматривается вместе с входными и выходными данными функции, то есть с потоком данных (data flow) в DO-178C. Типичная структурированная последовательность показана ниже:



В подписях рисунка надпись над фигурой показывает исполнителя соответствующего шага, а не отдельный объект потока. Эксперт означает эксперта по предметной области (Subject Matter Expert, SME); представитель УК означает представителя управления конфигурацией (Configuration Management Representative, CM Rep); УК означает управление конфигурацией (Configuration Management, CM); ед. конф. означает единицу конфигурации (Configuration Item, CI). В оригинале роли подписаны очень компактно и без дорожек ответственности, поэтому схема сама по себе читается хуже, чем должна.

В объектно-ориентированном проектировании ПО отдельные аспекты потока данных и потока управления инкапсулируются в объектах, как показано на этой диаграмме классов. Обратите внимание: низкоуровневые элементы потоков данных и управления здесь менее заметны, чем на структурной диаграмме выше.



Области, критичные по безопасности, включая авиацию, не любят риск: новые технологии считаются подозрительными, пока их безопасность не доказана. В критичном по безопасности ПО первостепенны детерминизм и верифицируемость. Многие годы сторонники традиционного подхода считали функциональную разработку ПО более детерминированной, чем объектно-ориентированную технологию: порядок выполнения структурированного ПО легко определить и повторить. На уровне модулей, то есть функций ПО и групп функций в одном файле, функциональное структурированное ПО также легче верифицировать: участки исходного кода можно напрямую трассировать к соответствующим требованиям низкого уровня к ПО (Low-Level Requirements, LLR) и тестировать последовательность за последовательностью. Поэтому функциональное

проектирование ПО десятилетиями надежно обеспечивало детерминизм и верифицируемость. При такой успешной истории зачем кому-либо понадобилась бы объектно-ориентированная технология с ее радикальной сменой парадигмы? Ответ прост: сама сущность *эволюции*...

Согласно дарвинизму, эволюция в природе происходит тогда, когда генетическое изменение дает преимущество для выживания. В технологии эволюция происходит сходным образом: изменение дает экономическое преимущество, а значит, помогает коммерческому выживанию. Переход ПО от функционального проектирования к объектно-ориентированной технологии произошел по простой причине: объектно-ориентированная технология все яснее давала два экономических преимущества перед функциональным проектированием: 1) лучшую способность управлять растущей сложностью ПО и 2) лучшую повторную используемость. Так и проросли семена эволюции авиационного ПО.

Понять необходимость объектно-ориентированной технологии – значит понять необходимость DO-332:

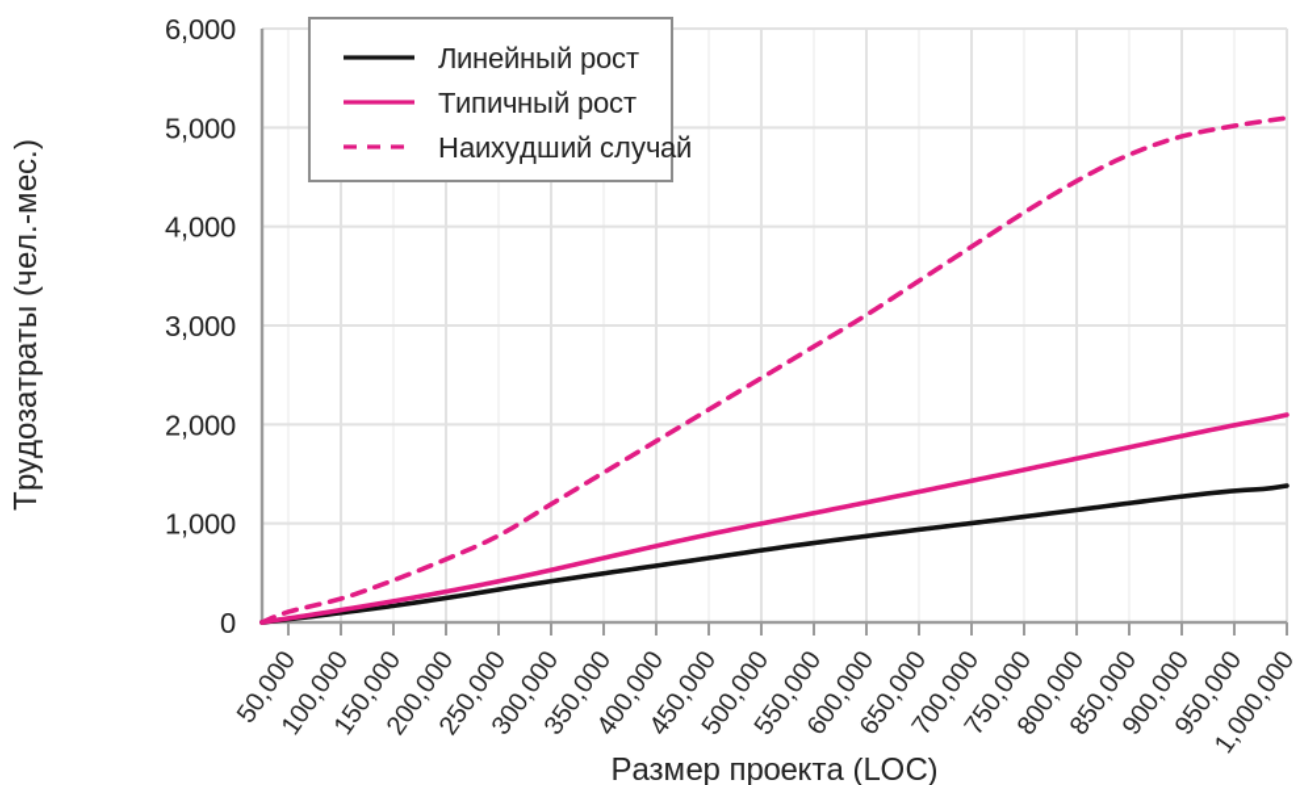
авиационное ПО, как и любое другое ПО, резко росло в размере, сложности и, следовательно, стоимости. Повышение безопасности означало расширение функциональности ПО, а это, в свою очередь, увеличивало его размер и сложность. Функциональное структурированное ПО вполне подходит и даже выгодно, когда функциональность проста. Но вычислительная мощность экспоненциально росла по закону Мура, и авиационные разработчики смогли использовать эту возросшую мощность.

В недавнем исследовании для одного авиационного заказчика автор проанализировал размер и стоимость авионического ПО по типам воздушных судов. Полученные оценки сведены в таблицу ниже. Следует отметить, что эти данные неформальны и недоказуемы: изготовители не раскрывают финансовые подробности, а значительная часть этого анализа и разработки выполнялась по соглашениям о неразглашении.

Тип воздушного судна	Приблизительное общее число строк ПО и логики	Средняя стоимость одной строки ПО и логики	Стоимость разработки ПО
Поршневая авиация общего назначения, 6+ мест	100 тыс. – 1,5 млн	\$260	\$26 млн – \$390 млн
Gulfstream G550	4,0 млн	\$330	\$1,32 млрд
Boeing 777 (1995)	4 млн	\$260	\$1,04 млрд
Boeing 787 (2012)	7 млн	\$370	\$2,59 млрд
JSF-35 (едининый ударный истребитель, Joint Strike Fighter, по состоянию на 2016 год)	15 млн	\$520	\$7,8 млрд (бортовая часть)

Физические и финансовые аспекты авионического ПО по репрезентативным типам воздушных судов

Следующий график, подготовленный доктором Барри Боэмом, автором конструктивной модели стоимости ПО (Constructive Cost Model, COCOMO) и профессором Университета Южной Калифорнии (University of Southern California, USC), где автор получил одну из своих магистерских степеней, показывает трудозатраты на разработку ПО в зависимости от размера проекта, измеренного в строках кода (Lines of Code, LOC). Этот график особенно важен, потому что более 90 % бортового ПО находится в системах, размер которых по отдельности лежит в диапазоне от 50 тыс. до 1 млн строк кода, то есть в пределах этого графика.



Очевидно, что размер, сложность и стоимость авиационного ПО неуклонно росли вместе с доступной вычислительной мощностью. Кроме того, распространение разных конфигураций систем повышало потребность в повторном использовании ПО. Полезные свойства структурированного ПО с точки зрения детерминизма и верифицируемости ослабевают по мере роста сложности системы, а повторное использование само по себе становится непростым: казалось бы, небольшие изменения логики могут потребовать большого анализа и переработки, чтобы сохранить правильную работу. Коммерческий мир за пределами критичного по безопасности ПО гораздо раньше оценил силу объектно-ориентированной технологии для реализации все более крупных и сложных систем с улучшенной повторной используемостью. Авиация охотно использует передовые технологии, но сторонится технологий на острие риска. В чистом виде объектно-ориентированная технология содержит ряд свойств, которые могут затуманивать детерминизм и усложнять верификацию. Поэтому к ней относились с подозрением. DO-178B и DO-278 появились до широкого рассмотрения объектно-ориентированной технологии и потому давали мало указаний. Однако такие языки, как Ada 95, уже содержали ключевые свойства объектно-ориентированной технологии, возможно, опередив свое время. Хотя почти нет сомнений, что объектно-ориентированная технология может улучшить управляемость ПО и его повторную используемость, у нее есть свойства, которые создают трудности для детерминизма и верифицируемости. DO-332 стал прямым ответом на эти трудности. Чтобы лучше понять проблемы объектно-ориентированной технологии для критичных по безопасности систем и рекомендации DO-332 по обеспечению соответствия, сначала нужно кратко разобраться в самой объектно-ориентированной технологии.

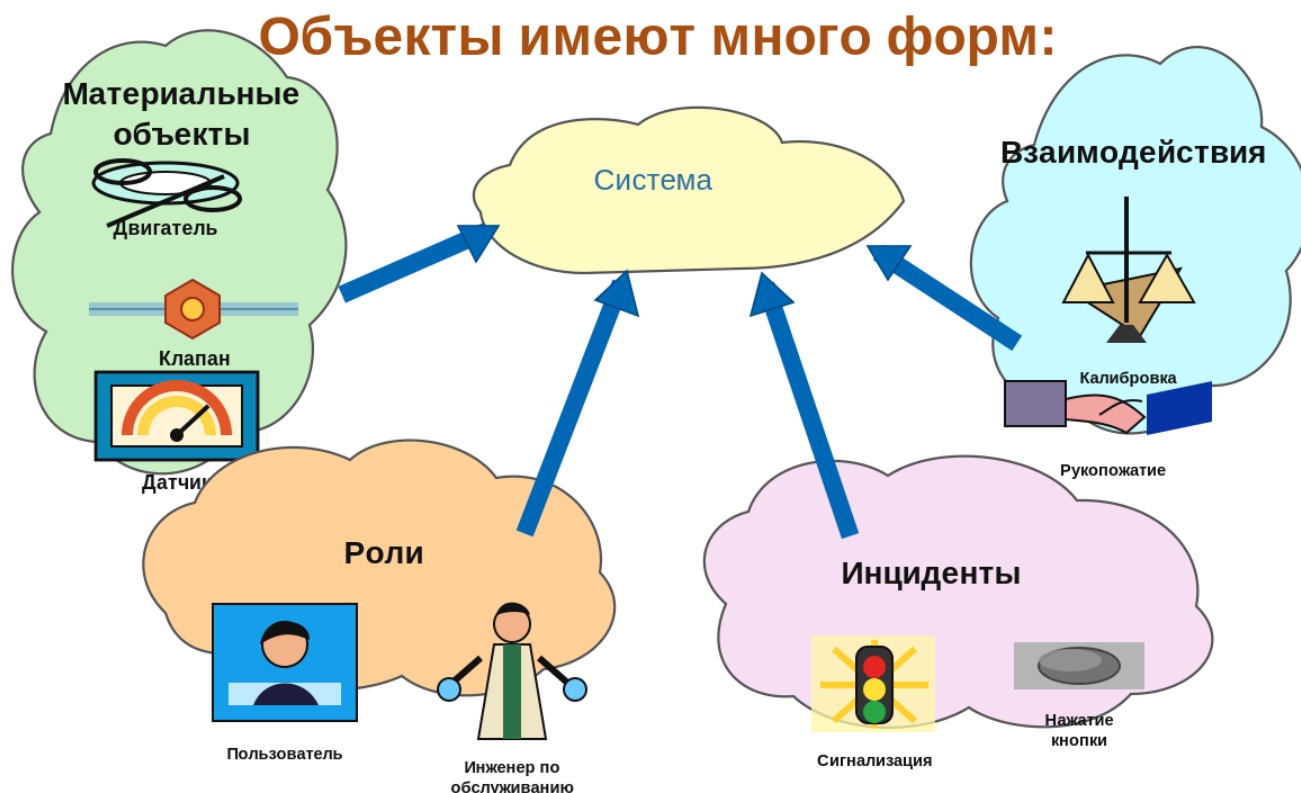
Предпосылки объектно-ориентированной технологии (ООТ)

В железный век вычислительной техники, то есть пятьдесят – двадцать пять лет назад, ПО писали вручную: продумывали последовательное выполнение инструкций, необходимых для достижения цели. Когда требовалась тесная поддержка аппаратуры, предпочтение отдавали языку ассемблера для более непосредственного управления на уровне центрального процессора (Central Processing Unit, CPU). В остальных случаях для научного программирования обычно использовали такие языки, как FORTRAN, Ada или С.

Авиационное ПО экспоненциально росло в 70-х и 80-х годах, поэтому основными языками стали Ada и С. Повышение повторной используемости и управляемости сложностью становилось все важнее, и грамотные разработчики применяли разные приемы для достижения этих целей: инкапсуляцию, абстрагирование аппаратуры, обертки и построение библиотек компонентов ПО с обобщенными и робастными интерфейсами. Хотя эти приемы улучшали повторную используемость и управление сложностью, коммерческие потребительские и финансовые секторы пошли гораздо дальше. Они полностью переосмыслили саму идею программирования как написания последовательных инструкций и вместо этого приняли объектно-ориентированное

программирование, ООП (Object-Oriented Programming, OOP), используя разные языки, специально разработанные для его поддержки. Мир критичных по безопасности систем двигался медленнее, потому что объектно-ориентированное программирование создавало трудности для верификации, а значит, и для сертификации. Чтобы понять, почему эти трудности возникали, нужно сначала понять основы объектно-ориентированного программирования.

Вместо того чтобы просто продумывать и писать, то есть кодировать, последовательные инструкции для компьютерной программы, разработчики объектно-ориентированного ПО мыслят объектами. Объект содержит инкапсулированные данные и процедуры, объединенные вместе и представляющие некоторую сущность. Объект – это структура данных, содержащая данные. Инструкции, то есть код, реализуются в процедурах, которые называются методами. У объекта есть интерфейсы, описывающие его взаимодействие внутри программы. Вместо мышления отдельными последовательными инструкциями разработчики объектно-ориентированного ПО работают на более высоком уровне: определяют объекты и взаимодействия, состоящие из групп инструкций, а не из отдельных последовательных инструкций. Методы объекта могут обращаться к данным внутри объекта и, возможно, обновлять их. Объекты имеют много форм, как показано ниже:



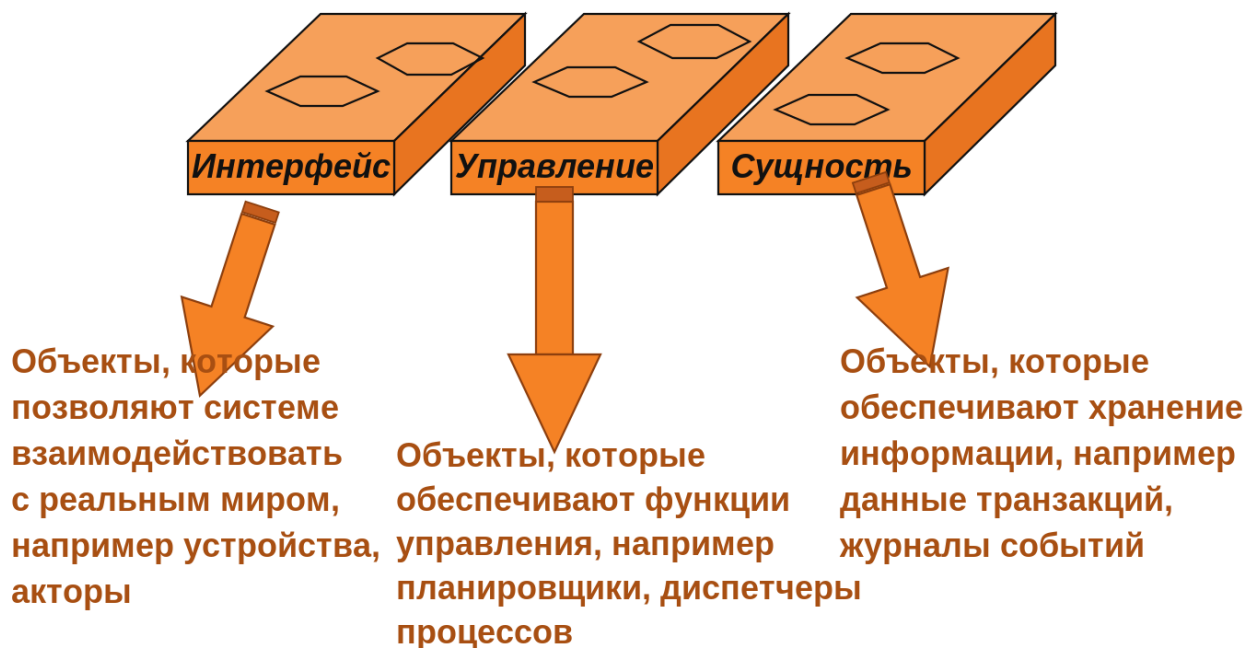
Объекты могут принимать разные формы, но в каждой из них объект сохраняет следующие свойства:

- является абстракцией понятия или предмета реального мира;
- имеет четкую границу;

- имеет уникальную идентичность;
- знает кое-что о самом себе;
- может выполнять действия над самим собой;
- может взаимодействовать с другими объектами.

Подобно людям, профессиям и даже воздушным судам, объекты могут сильно различаться по функциональности, возможностям и сложности. В самом простом виде есть три базовых типа объектов, показанные ниже:

Базовые типы объектов



Как видно, сами по себе объекты способны на многое и интересны. Но сами по себе они не слишком полезны. Рассмотрим авиационный двигатель: сам по себе он тоже способен на многое и представляет интерес. Однако двигатель становится по-настоящему мощным и полезным, когда объединяется с конструкцией воздушного судна, крыльями и системами управления. Так же объект становится мощным и полезным, когда используется в контексте объектно-ориентированного программирования. Базовые концепции объектно-ориентированного программирования – это возможности проектирования ПО, встроенные в язык программирования. В авиации и во многих современных критичных по безопасности системах таким языком обычно является C++. Эти особенности объектно-ориентированной технологии кратко показаны на рисунке ниже:

Ключевые особенности объектно-ориентированного ПО

- **Класс.** Группирование связанных структур данных с методами и функциями, работающими с этими данными.
- **Сокрытие информации.** Компоненты объекта могут быть открытыми или закрытыми; закрытые компоненты недоступны извне.
- **Наследование.** Подкласс наследует свойства родителя; собственные функции и данные могут расширять или переопределять их.
- **Полиморфизм.** Возможность заменять объект его подобъектами; может включать перегрузку.
- **Перегрузка.** Использование одного имени или символа для разных целей либо с разными аргументами.
- **Преобразование типов.** Преобразование представления исходного значения в другое целевое представление.
- **Управление динамической памятью.** Создание, выделение и освобождение памяти во время выполнения.
- **Виртуализация.** Абстрагирование ресурсов для эмуляции или имитации выполнения.

Что требуется для объектно-ориентированной технологии (ООТ) в авиации?

Значительная часть 150-страничного объема DO-332 объясняется его учебным характером: для единообразного обеспечения соответствия нужно задать основу объектно-ориентированной технологии и ее определения. DO-332 не претендует на роль полного учебника по объектно-ориентированной технологии, но рассматривает наиболее распространенные вопросы, связанные с надежностью и верификацией. Эти вопросы в основном относятся к тем аспектам, которые уникальны для объектно-ориентированной технологии и показаны выше. DO-332 задает процесс разработки ПО, параллельный процессу DO-178C, а именно требует письменные процессы применения объектно-ориентированной технологии и добавления связанных с ней аспектов к следующим уже обязательным мероприятиям по авиационному ПО:

- *Планы и стандарты*
- *Требования к ПО*
- *Проект ПО*
- *Код ПО*
- *Интеграция ПО*
- *Верификация ПО*

- *Управление конфигурацией*
- *Гарантия качества*
- *Сертификация*

Объектно-ориентированная технология в критичных по безопасности системах должна учитывать и снижать влияние тех потенциально рискованных аспектов, которые по природе присущи только ей. Что это за аспекты? Все просто: те, которые добавляют уязвимость к требуемым DO-178C трассируемости, согласованности, детерминизму, рассмотрением, структурному покрытию и тестированию.

“Уязвимость” – Потенциальная слабость, которая может снизить надежность системы.

Для работы с потенциальными уязвимостями объектно-ориентированной технологии DO-332 требует режима оценки, предотвращения и снижения последствий, ориентированного на конкретное применение этой технологии в каждом авиационном проекте. Поэтому проект должен определить, как именно объектно-ориентированная технология будет применяться в бортовом (DO-178C), наземном или космическом (DO-278A) исполнении ПО. Затем нужно выполнить анализ уязвимостей, специфичный для объектно-ориентированной технологии данного проекта. Проекты определяют свои соображения по объектно-ориентированной технологии в следующих материалах:

- планы по ПО: план сертификации ПО (Plan for Software Aspects of Certification, PSAC), план гарантии качества ПО (Software Quality Assurance Plan, SQAP), план управления конфигурацией ПО (Software Configuration Management Plan, SCMP), план разработки ПО (Software Development Plan, SDP) и/или план верификации ПО (Software Verification Plan, SVP);
- стандарты по ПО: стандарты требований, проектирования и/или кода, инженерные контрольные перечни и контрольные перечни гарантии качества.

Планы, стандарты и контрольные перечни каждого авиационного проекта должны соответствовать DO-332 и, следовательно, отражать результаты анализа уязвимостей, специфичного для этого проекта. Поэтому каждый проект должен рассмотреть, какими уязвимостями может обладать применяемая в нем объектно-ориентированная технология. Каждый проект должен проанализировать свою среду разработки ПО и жизненный цикл, чтобы выявить потенциальные уязвимости объектно-ориентированной технологии. Для тех уязвимостей, которые потенциально могут существовать, нужно определить стратегию снижения последствий и последующей верификации, а затем доказуемо выполнить эту стратегию. DO-332 описывает многие распространенные уязвимости объектно-ориентированной технологии и основные шаги по их снижению, которые кратко приведены ниже.

Основные аспекты объектно-ориентированной технологии и связанные соображения по уязвимостям

- **Наследование: меры снижения уязвимостей.**
 - Обеспечивать принцип подстановки Лисков.
 - Проверять неиспользуемый код.
 - Избегать множественного наследования.
- **Параметрический полиморфизм: меры снижения уязвимостей.**
 - Поддерживать согласованность данных.
 - Обеспечивать трассируемость исходного и двоичного кода.
 - Проверять все типы.
- **Перегрузка: меры снижения уязвимостей.**
 - Избегать неоднозначности и конфликтов имен.
 - Использовать явные, а не неявные преобразования типов.
- **Преобразование типов: меры снижения уязвимостей.**
 - Использовать явные правила кодирования для приведений и преобразований.
- **Управление исключениями: меры снижения уязвимостей.**
 - Избегать непроверяемых исключений.
 - Локализовать обработку исключений.
- **Управление динамической памятью: меры снижения уязвимостей.**
 - Применять управление объектами и сборку мусора.
 - Избегать нехватки и исчерпания памяти.
- **Виртуализация: меры снижения уязвимостей.**
 - Проверять каждый слой виртуализации.
 - Обеспечивать трассировку всех интерпретаций.

У объектно-ориентированной технологии очевидно много преимуществ, и ее применение растет в областях критичного по безопасности ПО. Причины ее первоначально медленного внедрения в авиации во многом связаны с уязвимостями, перечисленными выше. Эти уязвимости снижают, ограничивая соответствующие возможности объектно-ориентированной технологии, а затем верифицируя соблюдение таких ограничений через рассмотрения требований, проекта и кода на соответствие связанным стандартам. Но даже если бы эти ограничения могли полностью устранить уязвимости, а они не могут, все равно остается несколько связанных с объектно-ориентированной технологией вопросов, которые нужно рассмотреть. Они описаны ниже.

Проблемы и уязвимости объектно-ориентированной технологии (ООТ) в критичном по безопасности ПО

Для критичного по безопасности ПО должно быть доказано, что оно удовлетворяет заданным целям. Авиационное ПО, как и в большинстве областей критичных по безопасности систем, получает уровень гарантии для компонентов ПО в зависимости от потенциального вклада этого ПО в последствия для безопасности. Для более высоких уровней гарантии, то есть уровней гарантии разработки (Development Assurance Level, DAL) А-С для авионического ПО, уровней гарантии разработки А-В для сложной авионической электронной аппаратуры и уровней гарантии (Assurance Level, AL) 1-3 для авиационных систем связи, навигации, наблюдения и организации воздушного движения (Communication, Navigation, Surveillance / Air Traffic Management, CNS/ATM), эти цели требуют оценки фактического проекта логики и реализации. Такую оценку проекта и реализации называют методом белого ящика (*white-box*), потому что оценка должна “видеть внутри коробки” проекта и реализации. Напротив, более низкие уровни гарантии не требуют оценки методом белого ящика, но, как и более высокие уровни, требуют оценки методом черного ящика (*black-box*).

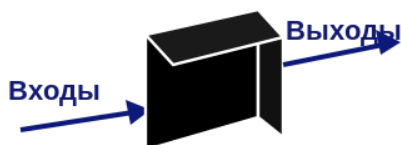
На рисунке ниже показано различие между методом черного ящика и методом белого ящика.

Черный ящик и белый ящик

В чем различие?

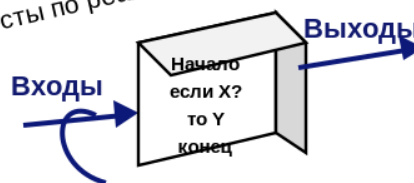
Метод черного ящика:

- Не зависит от реализации
- Нельзя «заглянуть внутрь»
- Тесты по требованиям



Метод белого ящика:

- Зависит от реализации
- Анализ/использование реализации
- Нужно «видеть внутри»
- Тесты по реализации



Верификация методом черного ящика и методом белого ящика

Многочисленные преимущества объектно-ориентированной технологии не даются без риска. На уровне метода черного ящика различия между объектно-ориентированной технологией и функциональным ПО невелики. Они в основном относятся к требованиям к ПО, которые могут подразумевать предпочтительность объектно-ориентированного

проекта, например для сетей, связи и видео с сообщениями и пакетами, а также для динамической или асинхронной обработки с высокой пропускной способностью. Однако на уровне метода белого ящика отличия объектно-ориентированной технологии от функционального ПО требуют дополнительного рассмотрения следующих вопросов, для которых DO-332 указывает конкретные мероприятия.

Ключевые вопросы объектно-ориентированной технологии и соображения по их рассмотрению

• Трассируемость.

- Должно учитываться наследование, включая все созданные экземпляры классов и подклассов.
- Должна сохраняться трассируемость в моделях, особенно в модели проекта.
- Для уровня гарантии разработки (Development Assurance Level, DAL) А учитывать корреляцию исходного и объектного кода для всех элементов объектно-ориентированной технологии (Object-Oriented Technology, OOT): конструкторов, деструкторов, преобразований типов и кода, созданного компилятором.

• Анализ структурного покрытия.

- По тестированию на основе требований оценить покрытие и верифицировать не менее одного экземпляра на точку вызова.
- Показать покрытие всех унаследованных и явных методов для каждого класса.
- Рассмотреть связность по управлению и данным на уровнях проекта и кода объектно-ориентированной технологии.

• Разработка на основе компонентов.

- "Святой Грааль" ПО – повторно используемые компоненты; требуется полная верификация каждого повторного применения, включая трассировку и деактивированный код.
- Доказать анализ связности по управлению каждого компонента в интегрированной системе.
- Верифицировать управление ошибками и ресурсами в целевой системе.

• Анализ ресурсов.

- Верифицировать использование стека и кучи: динамическое выделение и освобождение памяти.
- Определить наихудшее время выполнения (Worst-Case Execution Time, WCET) и верифицировать его достаточность.
- Подтвердить детерминированность операций с памятью тестами, доказывающими наихудшее время выполнения.
- Подтвердить уникальность ссылок на объекты и атомарность всех перемещений.

Заключение по объектно-ориентированной технологии (ООТ) в DO-332

Объектно-ориентированная технология дает много преимуществ, особенно для все более сложных и постоянно развивающихся авиационных программных систем. Она позволяет значительно улучшить повторную используемость ПО, а это святой Грааль разработки ПО, особенно в авиации. По сути, при полном использовании ранее сертифицированных компонентов ПО нужно повторять только повторное тестирование и повторные рассмотрения тестирования; объектно-ориентированная технология существенно помогает в этом. Возможно, самое большое препятствие для повторного использования ПО – политическое, а не техническое: все знают, что создание повторно используемых компонентов ПО экономит огромные средства в будущих проектах, но такое создание повторной используемости увеличивает начальные затраты на разработку. Первый проект оплачивает повышенные затраты на закладывание повторной используемости в проект, а последующие проекты получают выгоду. Разумным компаниям стоило бы уделять больше внимания повторной используемости и ее долгосрочным стоимостным преимуществам; объектно-ориентированная технология – один из секретов достижения этого. При всей пользе она имеет уязвимости, которые могут создавать риски. Аналогия со здоровьем человека здесь вполне уместна: людям почти всегда полезны кардиотренировки и силовая подготовка. Но бегуны-любители и случайные тяжелоатлеты нередко оказываются по субботам в приемных отделениях больниц, где узнают, что более ограниченные режимы тренировок дают лучшие и более безопасные результаты. Для объектно-ориентированной технологии рекомендуемые лучшие практики кратко приведены ниже:

Обеспечьте, чтобы каждый компонент, подкласс и интерфейс имел собственные:

- *Трассируемость требований*
- *Тестирование: функциональное, на робастность и структурное*
- *Шаблоны должны порождать экземпляры и тестироваться для каждого типа аргумента, если только не получается идентичный двоичный результат (класс эквивалентности)*
- *Вложенные шаблоны и дружественные классы запрещены для уровней А, В и С.*
- *Убедитесь, что связность по данным и связность по управлению учитывают вашу реализацию на C++ для уровней А, В и С.*
- *Прочитайте Консультативный циркуляр (Advisory Circular, AC) 20-148 и применяйте изложенные в нем практики повторного использования ПО даже тогда, когда не запрашиваете у сертифицирующего органа официальную классификацию повторно используемого программного компонента (Reusable Software Component, RSC).*

Для безопасности и “здоровья” ПО столь же полезным окажется четкое понимание объектно-ориентированной технологии и формальных ограничений при разработке ПО. Любое обучение объектно-ориентированной технологии начинайте с внимательного чтения DO-332: это превосходное введение в безопасную разработку.

DO-330. Квалификация инструмента (TQ)

Автор: Вэнс Хилдерман

Рецензенты:

Буньямин Кошкун, руководитель проекта, Milsoft Inc.; Дэйв Делориа, директор по инженерным работам, Crane Aero; Роджер Плиске, Rohde & Schwarz

Авиационные инженерные “инструменты”: обзор

Инженерные “инструменты” для систем, ПО и аппаратуры – это компьютерные программы, которые помогают инженерам создавать, анализировать, верифицировать, отслеживать, изменять, производить или специфицировать разрабатываемое прикладное ПО. Такие инструменты используются почти с самого начала эпохи вычислительной техники. Да, в инженерной работе есть и более материальные инструменты: осциллографы и предметы из настоящего ящика техника по обслуживанию или монтажу. Это тоже инструменты, но не в том смысле, который вкладывается в это слово в авиационной инженерии. В DO-330 термин “инструмент” буквально относится к программам или программной логике, применяемым при инженерной разработке или верификации авиационных систем.

Инструменты повышают эффективность процесса разработки, автоматизируя рутинные или сложные операции. Кроме того, они приближают уровень абстракции к тому, как разработчик понимает задачу. Но всегда ли инструментам можно доверять, или в некоторых случаях они должны проходить формальную квалификацию? Ответ на этот важный вопрос приведен ниже вместе с подробностями выполнения такой квалификации, когда она действительно нужна. Это не вопрос из серии “черное или белое”, а прагматичная оценка: в какой степени можно доверять инструменту и при каких условиях инструменты необходимо формально квалифицировать.

Современные изделия высокой надежности используют инструменты разработки и верификации во множестве применений, критически важных с точки зрения безопасности; без таких инструментов инженерный процесс разработки практически невыполним. Эти инструменты могут устранять, сокращать или автоматизировать процессы, которые обеспечивают корректность критически важного с точки зрения безопасности приложения. Системы для авиации, медицины, железных дорог, космоса, автомобильной промышленности и военной отрасли разрабатываются с применением инструментов, которые сами могут способствовать ошибочной работе и привести к недопустимому поведению приложения. Среда разработки может влиять на проект и

поведение изделия, и это влияние нужно учитывать. Кроме того, инструмент, помогающий верификации, может работать некорректно и тем самым оставить ошибку в изделии необнаруженной.

В авиационной отрасли потенциально отрицательное влияние инженерных инструментов на авионические изделия должно быть снижено. Для программ разработки ПО это регулируется документами Радиотехнической комиссии по аэронавтике (Radio Technical Commission for Aeronautics, RTCA) DO-178C/ED-12C, а для программ разработки аппаратуры – RTCA/DO-254/ED-80. Наземные системы связи, навигации, наблюдения и организации воздушного движения (Communication, Navigation, Surveillance / Air Traffic Management, CNS/ATM) имеют сходные потребности в квалификации инструментов. Что общего у этих авиационных систем? Все они опираются на руководство DO-330 “Вопросы квалификации программных инструментов” (Software Tool Qualification Considerations) по квалификации инструмента. При этом во многих случаях DO-330 фактически не требует квалифицировать программный инструмент, если выходные данные этого инструмента верифицируются иным способом. Общие сведения о квалификации инструмента по DO-330 приведены на следующем рисунке и подробно рассматриваются далее в этой главе:

Общие сведения о квалификации инструмента по DO-330

- Инструменты – это нелетное ПО.
- Они автоматизируют, дополняют или заменяют этапы сертификации.
- Инструменты относятся к особой области, но не являются бортовым ПО.
- Разработчики и пользователи инструментов часто различаются.
- Строгость квалификации зависит от применения инструмента и его влияния.
- DO-330 применим и к другим областям, а не только к DO-178C.

Что такое инструмент?

Словарь Вебстера определяет инструмент как приспособление; любое средство достижения цели; нечто, используемое при выполнении операции; то, что считается необходимым для выполнения работы или профессии; либо человека, которого использует другой человек или которым он манипулирует. Последнее – тема для другого разговора, но аналогия занятная. RTCA/DO-178C, RTCA/DO-254 и приказ Федерального управления гражданской авиации США (Federal Aviation Administration, FAA) 8110.49 определяют инструмент как компьютерную программу, используемую для разработки, тестирования, анализа, создания или модификации другой программы либо ее документации. Следовательно, для авионики инструмент – это само ПО, применяемое где-либо в жизненном цикле авионических систем. Рассмотрите распространенные типы инструментов разработки ПО на рисунке ниже и задайте себе вопрос, нужно ли их квалифицировать. Ответ будет дан в следующем разделе.

Рисунок 1: Распространенные типы инструментов программной инженерии

Инструменты разработки ПО:

- компиляторы;
- компоновщики;
- инструменты моделирования;
- генераторы кода.

Инструменты верификации ПО:

- статический анализ кода;
- выполнение тестов;
- структурное покрытие;
- проверка прохождения тестов.

Каждый из инструментов на рисунке выше играет важную роль в авионической инженерии, и каждый доступен как готовое коммерческое ПО (COTS), которое можно напрямую приобрести у одного из десятков поставщиков программных инструментов. В DO-178B инструменты просто делились на “инструменты разработки” и “инструменты верификации”. Однако DO-178C отказывается от такой простой классификации: технический прогресс позволил создавать гибридные инструменты, которые выполняют верификацию и одновременно сокращают последующие действия разработки. Ниже это объясняется через “критерии” инструмента.

Инструменты, используемые в инженерной работе, встречаются на всех этапах проекта: спецификация требований, проект ПО и код, интеграция, управление конфигурацией и верификация. В авиационной отрасли все еще можно разработать приложение, критичное для безопасности, используя только инструменты трансляции и сборки, такие как компилятор, ассемблер и компоновщик. Но с учетом сложности и масштаба современных электронных систем и авионики это становится все менее вероятным.

Когда необходима квалификация инструмента (TQ)?

Квалификация инструмента требуется всякий раз, когда процессы гарантии разработки, описанные в RTCA/DO-178C или RTCA/DO-254, устраняются, сокращаются или автоматизируются за счет использования инструмента, если только выходные данные этого инструмента не верифицируются. Верификация выходных данных инструмента должна выполняться через процесс верификации, определенный в разделе 6 RTCA/DO-178C. Важно помнить: в разработке авионики термин “верификация” имеет специальное значение. Следующая формула не является официальной политикой FAA или Агентства Европейского союза по авиационной безопасности (European Union Aviation Safety Agency, EASA); автор сформулировал ее сам и называет “уравнением верификации DO-XXX”.



Рисунок 2: Уравнение верификации DO-XXX

По структуре DO-330 построен по образцу DO-178C, поскольку был выпущен почти одновременно с ним. DO-330 включает три сходных ключевых процесса: планирование, разработку и интегральную корректность, как показано на рисунке ниже:

Схематическое представление структуры DO-330

1. Введение.
2. Назначение квалификации инструмента.
3. Характеристики квалификации инструмента.
4. **Планирование:** планирование квалификации инструмента.
5. **Разработка:** процесс разработки инструмента.
6. Верификация инструмента.
7. Управление конфигурацией инструмента.
8. Обеспечение качества инструмента.
9. Взаимодействие по квалификации.
10. Данные квалификации инструмента.
11. Дополнительные соображения.

Приложение А: таблицы целей квалификации инструмента T-0 – T-10.

Разделы 6-11 и приложение А относятся к интегральным процессам и обеспечению корректности.

Квалификация инструмента (TQ) в DO-178B и DO-178C

В прежнем DO-178B, который, конечно, вытеснен DO-178C, квалификация инструмента рассматривалась непосредственно в самом DO-178B и уточнялась широко применяемым приказом FAA 8110.49. Инструменты относились всего к одной из двух категорий:

1. **Инструменты разработки:** *могут внести ошибку в эксплуатационное бортовое ПО; или*
2. **Инструменты верификации:** *не способны внести ошибку, но потенциально могут не обнаружить ошибку в полетном ПО.*

Однако DO-178B вышел в 1992 году, на раннем этапе развития передовых методов разработки ПО, еще до появления связанных руководств для сложной электронной аппаратуры (DO-254) и систем CNS/ATM (DO-278A). Руководство по квалификации инструмента в DO-178B занимало меньше четырех страниц и часто действительно оказывалось слишком кратким. Поэтому потребовалось новое руководство по квалификации авиационных программных инструментов. Основные причины показаны на следующем рисунке:

Рисунок 3: Причины появления DO-330 как нового самостоятельного руководства по квалификации инструмента

1. **Согласованное руководство по квалификации инструментов** для DO-178C, DO-254, DO-278A и рекомендуемой аэрокосмической практики ARP4754A (Aerospace Recommended Practice, ARP).
2. **Технический прогресс в возможностях инструментов:** моделирование и проектирование, анализ, интеллектуальное тестирование.
3. **Учет различных преимуществ инструментов:** преимущества для разработчика инструмента и для пользователя инструмента.
4. **Повышенная строгость квалификации** для более высоких уровней гарантии разработки (Development Assurance Level, DAL).

Как показано выше, для введения DO-330 “Вопросы квалификации программных инструментов” (Software Tool Qualification Considerations) было четыре ключевые причины: требовалось собрать в одном документе необходимые дополнительные сведения по квалификации инструмента. Определение того, нужно ли квалифицировать конкретный инструмент, выполняется через оценку ответов на три вопроса независимо от категории инструмента.

1. Может ли инструмент в рамках предполагаемого использования внести ошибку в бортовое ПО или аппаратуру либо не обнаружить уже существующую ошибку в ПО или аппаратуре?
2. Останутся ли выходные данные инструмента без верификации или подтверждения другими действиями по верификации, например действиями из раздела 6 DO-178C для ПО?
3. Будут ли выходные данные инструмента использоваться для выполнения цели или для замены цели RTCA/DO-178C, DO-254, DO-278A и т. д.?

Если на все три вопроса ответ положительный, то инструмент, скорее всего, потребует квалифицировать. На рисунке ниже эта логика представлена в виде классической блок-схемы. Следует отметить, что ответы на первый и третий вопросы почти всегда положительные, поэтому реальная необходимость квалификации обычно сводится к одному вопросу: будут ли выходные данные инструмента верифицированы? Если ответ

отрицательный, квалификация почти всегда требуется. Простая блок-схема этих вопросов приведена ниже на рисунке 4. Если говорить прямо, большинство инструментов либо могут внести ошибку, либо могут не обнаружить ошибку; кроме того, большинство инструментов устраняют, сокращают или автоматизируют процессы разработки и верификации авионики. Поэтому главный вопрос при решении о квалификации инструмента звучит так: верифицируются ли выходные данные инструмента иным образом? Если выходные данные инструмента, применяемого по DO-178C, DO-254 или DO-278A, не верифицируются, такой инструмент почти всегда должен быть квалифицирован. После того как установлено, что инструмент нужно квалифицировать, для конкретного применения определяют его уровень квалификации инструмента (Tool Qualification Level, TQL), затем определяют применимые к данному экземпляру инструмента цели DO-330 и выполняют их.

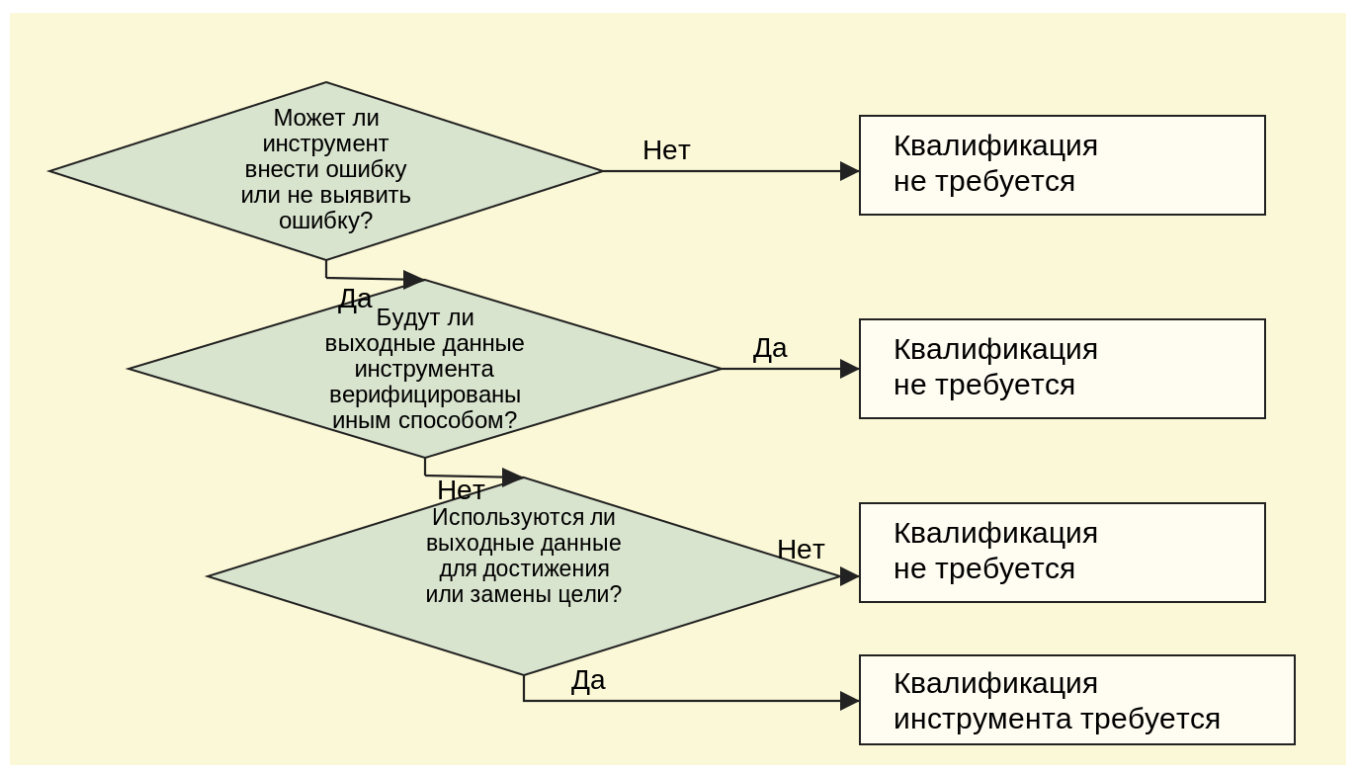


Рисунок 4: Упрощенная блок-схема для определения необходимости квалификации инструмента

Зачем нужна квалификация инструмента (TQ)?

Полетная аппаратура, ПО и системы обычно сертифицируются. Инструменты же используются при разработке и/или верификации, а сам инструмент обычно не летает и не исполняется на борту воздушного судна в полете. Однако на инструмент возлагается получение свидетельств и выходных данных, которые удовлетворяют сертификационным целям. Поэтому необходимо сформировать доверие и доказать, что инструмент обеспечивает по крайней мере эквивалентную гарантию тем сертификационным процессам, которые он устраняет, сокращает или автоматизирует. Надежность применяемого инструмента должна быть установлена. Установление надежности

инструмента и уверенности в том, что он обеспечивает как минимум эквивалентный процесс гарантии разработки для требуемого уровня, достигается через процесс квалификации инструмента. Первый шаг этого процесса – установить, нужна ли квалификация, как описано выше.

Решение о необходимости квалификации инструмента и его обоснование, независимо от того, требуется квалификация или нет, следует согласовать с сертифицирующим органом как можно раньше и отразить в документах сертификационного планирования. Оценку и анализ инструмента нужно выполнить на этапе планирования проекта, до перехода к разработке и верификации. Если установлено, что квалификация инструмента не требуется, такое согласие следует получить рано, чтобы избежать проблем позднее. Разработка может продолжаться до завершения квалификации инструмента; однако при разработке важно учитывать возможность квалификации, чтобы при необходимости ее можно было выполнить. За десятилетия автор видел множество проектов, где предполагалось, что квалификация инструмента не нужна. Итоги были печальными, когда позднее использование выходных данных такого инструмента запрещалось, потому что сам инструмент уже нельзя было квалифицировать.

DO-330 предусматривает, что действия по квалификации инструмента напрямую связаны с потенциальным влиянием инструмента. Это влияние определяется категорией инструмента и уровнем гарантии разработки (Development Assurance Level, DAL), к которому инструмент применяется. Поэтому DO-330 вводит пять уровней квалификации инструмента на основе трех критериев инструмента.

Рисунок 5: Уровни квалификации инструмента на основе уровня гарантии разработки и критериев инструмента

Уровень гарантии разработки ПО (DAL, уровень критичности)	Критерий инструмента 1	Критерий инструмента 2	Критерий инструмента 3
Уровень А	TQL-1	TQL-4	TQL-5
Уровень В	TQL-2	TQL-4	TQL-5
Уровень С	TQL-3	TQL-5	TQL-5
Уровень D	TQL-4	TQL-5	TQL-5

Существует три критерия инструмента. Иными словами, использование инструмента оценивается как относящееся к одной из трех следующих категорий:

- Критерий 1:** инструмент, выходные данные которого являются частью бортового ПО и поэтому могут внести ошибку.

Пример: инструмент генерации кода, который автоматически создает исходный код по моделям.

- Критерий 2:** инструмент, который автоматизирует процессы верификации и поэтому может не обнаружить ошибку, причем его выходные данные используются как обоснование для устранения или сокращения:

- а. *процессов верификации, отличных от тех, которые автоматизирует сам инструмент; или*
- б. *процессов разработки, которые могут влиять на бортовое ПО.*

Пример: инструмент проверки моделей, который верифицирует полноту и одновременно проверяет покрытие.

1. Критерий 3: *инструмент, который в рамках предполагаемого использования может не обнаружить ошибку.*

Пример: инструмент анализа структурного покрытия, оценивающий покрытие кода.

Как определяется TQL? Рассмотрим предыдущий рисунок. Если выходные данные инструмента, например генератора кода, входят в состав ПО с DAL B, то это инструмент критерия 1 с TQL 2. С другой стороны, если инструмент является только инструментом верификации структурного покрытия и используется при любом DAL, то это инструмент критерия 3 с TQL 5.

Какие инструменты требуют квалификации (TQ)?

Не все инструменты требуют квалификации. С помощью трех вопросов, рассмотренных выше, достаточно легко установить, какие инструменты будут или не будут требовать квалификации. Инструменты, которые обычно относятся к управлению требованиями, управлению конфигурацией и данными, а также управлению качеством, как правило, можно исключить из процесса квалификации инструмента. Почему? Такие инструменты обычно не выдают выходные данные, которые выполняют цель приложения или заменяют такую цель, либо выходные данные инструмента верифицируются другим действием по верификации ниже по процессу. Однако оценка инструмента все равно должна быть выполнена; именно она устанавливает, нужна квалификация или нет. Инструменты, которые обычно требуют квалификации, достаточно ясно попадают под критерий 1, 2 или 3. Важно на раннем этапе определить, требуется ли квалификация инструмента, и если да, то какой TQL ему соответствует. Рекомендация: даже если оценка инструмента показывает, что квалификация не требуется, укажите инструмент в документах сертификационного планирования и приведите обоснование, почему квалификация не нужна. Лучше быть честным и открытым сразу, чем ловить проблемы позднее по проекту.

Компиляторы, ассемблеры и компоновщики обычно являются инструментами критерия 1, но их выходные данные часто проверяются другим действием по верификации, например рассмотрением или тестированием. Поэтому обычно они не требуют квалификации. К примерам инструментов критерия 1, которые могут требовать квалификации, относятся: инструменты проектирования, генерирующие исходный код, то есть генераторы кода; инструменты реализации, создающие исполняемый код или его

машинно-читаемое представление; инструменты, создающие представления кода, либо имитаторы, которые сами не являются фактическим кодом; а также инструменты двоичной трансляции, например кросс-компиляторы или генераторы форматов.

Примечание переводчика. В оригинале сказано: “implementation tools that produce executable code representations; code representations or simulation tools (i.e. not actual)”. Под “представлением исполняемого кода” здесь понимается не текст исходного кода, а результат, который описывает или содержит будущий исполняемый образ в форме, пригодной для дальнейшей сборки, загрузки или анализа. Пример: генератор загрузочного HEX/S-record-файла, ELF-образа или иного формата памяти по результатам компоновки; такой файл еще может быть входом следующего шага, но ошибка в нем способна попасть в исполняемый объектный код.

Примеры инструментов критериев 2 и 3, которые могут требовать квалификации: инструменты, автоматизирующие рассмотрения кода и проекта на соответствие стандартам; инструменты, генерирующие тестовые примеры и/или процедуры по требованиям; инструменты, определяющие статус “пройдено/не пройдено”; инструменты, отслеживающие и выдающие результаты структурного покрытия; а также инструменты, определяющие результаты покрытия требований.

Сам авионический код, библиотеки компилятора и операционные системы реального времени, ОСПВ (Real-Time Operating System, RTOS), не считаются инструментами, поскольку они входят в состав фактического исполняемого ПО или аппаратуры. Они верифицируются процессом гарантии разработки и требуют не квалификации инструмента, а полной сертификации “полетного ПО” по DO-178C.

Жизненный цикл инструментов, прошедших квалификацию (TQ)

Быстрый вопрос: можно ли встроить качество в изделие после того, как оно уже разработано? Конечно нет. Настоящее качество изделия опирается на качественное планирование, реализацию по плану и оценку реализации вместе с поддерживающими процессами. Основные шаги квалификации инструмента показаны ниже в типичной последовательности:

1. Определить, требуется ли квалификация инструмента.
2. Определить, какой критерий квалификации инструмента применим.
3. Определить, какой уровень квалификации инструмента (TQL) применим.
4. Определить применимые цели и данные жизненного цикла для каждого TQL.
5. Выполнить процессы жизненного цикла инструмента:
 - a. процесс планирования;
 - b. процесс требований;
 - c. процесс проектирования;
 - d. процесс кодирования;
 - e. процесс эксплуатационной интеграции;

- f. процесс верификации;
- g. процесс управления конфигурацией;
- h. процесс обеспечения качества;
- i. процесс взаимодействия с сертифицирующим органом.

Типовая последовательность квалификации инструмента

Так же как DO-178C требует процессов жизненного цикла для авионического ПО, DO-330 определяет жизненный цикл для квалифицированных инструментов, показанный на следующем рисунке. Жизненный цикл квалификации инструмента состоит из трех ключевых действий: 1) планирование инструмента, 2) разработка инструмента и 3) верификация инструмента. Эти действия должны выполняться последовательно: сначала планирование, затем разработка и наконец верификация. При этом в фоновом режиме на протяжении каждого из этих действий постоянно выполняются соответствующие интегральные процессы: управление конфигурацией инструмента, обеспечение качества инструмента и взаимодействие по квалификации инструмента.

Рисунок 6: Жизненный цикл квалифицированных инструментов по DO-330

1. **Планирование квалификации инструмента** (DO-330, раздел 4): планы, контрольные перечни и процессы.
2. **Разработка инструмента** (DO-330, раздел 5): эксплуатационные требования к инструменту, разработка инструмента (требования, проектирование, кодирование, интеграция) и эксплуатационная интеграция.
3. **Верификация инструмента** (DO-330, раздел 6): рассмотрения, тестирование и анализ инструмента, а также эксплуатационная валидация и верификация инструмента.

Интегральные процессы выполняются в фоне: управление конфигурацией, обеспечение качества и взаимодействие по квалификации инструмента.

Назначение уровня квалификации инструмента (TQL)

Итак, все инструменты идентифицированы, определено, какие из них должны быть квалифицированы и по какому критерию, и можно начинать квалификацию с планирования. Для планировщика или организации, отвечающей за планирование, критически важно хорошо знать процессы жизненного цикла организации или проекта и их соответствие целям DO-178C/DO-254, особенно если несколько инструментов вносят вклад в одну цель. Но что фактически требуется для самой квалификации инструмента? Это полностью зависит от TQL инструмента. Пять разных TQL нужны по простой причине: потенциальное неблагоприятное влияние неправильного использования инструмента или его выходных данных сильно различается от уровня к уровню. Проблемы инструмента TQL 1 обычно опаснее проблем инструмента TQL 5. Поэтому инструменты TQL 1 требуют наибольшей строгости квалификации и наибольшего

количества артефактов, а инструменты TQL 5 – наименьших. Если конкретный инструмент используется как инструмент TQL 3, можно ли квалифицировать его на более высокий уровень, например TQL 2 или TQL 1?

Да, конечно. Всегда можно сделать больше работы, чем требуется, особенно если бюджет и график не ограничены. Но, поскольку читатель разумен, он не станет этого делать, если только нет достаточно уверенности, что в следующем проекте потребуется доказать более высокий TQL. Можно ли квалифицировать инструмент на более низкий уровень, например TQL 4 или TQL 5? Разумеется, нет: на этих более низких уровнях, то есть при большем номере TQL, меньше целей квалификации. Рекомендация: разработка авионики и без того чрезвычайно дорога и трудоемка. Избегайте лишней работы и квалифицируйте инструмент на минимальный требуемый TQL, если только вы не уверены, что будете повторно использовать этот инструмент в другом проекте, где потребуется более высокий TQL.

Планирование квалификации инструмента (TQ)

Теперь TQL формализован, сертифицирующий орган формально одобрил ваши планы или вы обоснованно уверены, что он их одобрит, и можно начинать собственно квалификацию инструмента. Требуемые цели квалификации инструмента, которые должны быть выполнены, подробно приведены в десяти таблицах приложения А в конце DO-330. Эти цели зависят от TQL, как показано на следующем рисунке: обратите внимание, что требуемые цели квалификации инструмента увеличиваются по мере перехода от наименее строгого TQL 5 к наиболее строгому TQL 1.

Рисунок 7: ключевые цели и данные квалификации инструмента по уровню квалификации инструмента (Tool Qualification Level, TQL)

Инструменты TQL 5:

1. Охват плана сертификации ПО (Plan for Software Aspects of Certification, PSAC).
2. Эксплуатационные требования.
3. Идентификация и управление конфигурацией.
4. Базовая гарантия качества.
5. Верификация требований.

Инструменты TQL 4: как выше, плюс:

1. Управление эксплуатационными требованиями.
2. Детали планирования по пяти планам.
3. Процессы разработки и интеграции.
4. Более строгое управление конфигурацией.
5. Робастность тестов.

Инструменты TQL 3: как выше, плюс:

1. Стандарты требований, проекта и кода.
2. Критерии перехода.
3. Среда разработки.
4. Требования низкого уровня.
5. Покрытие операторов.
6. Связность данных и управления.

Инструменты TQL 2: как выше, плюс:

1. Независимость.
2. Анализ внешнего ввода/вывода.
3. Верификация исходного кода инструмента.
4. Верификация требований низкого уровня.
5. Покрытие решений.

Инструменты TQL 1: как выше, плюс:

1. Повышенная независимость.
2. Верификация внешних компонентов.
3. Модифицированное покрытие условий/решений (Modified Condition/Decision Coverage, MC/DC).

Первый шаг планирования квалификации программного инструмента – убедиться, что такая квалификация действительно нужна, как объяснялось выше. Если она нужна, рисунок выше обобщает ключевые цели и данные квалификации. Прежде всего важно понять, какие цели потребуются выполнить в зависимости от TQL вашего инструмента.

DO-330 перечисляет каждую конкретную цель по TQL, а на рисунке показаны ключевые цели. Еще раз: эти цели квалификации инструмента накапливаются. По мере роста строгости, от TQL 5 к TQL 1, требуются дополнительные цели. Может ли сегодняшний инструмент TQL 5 завтра стать инструментом TQL 4, а сегодняшний инструмент TQL 3 – инструментом TQL 2? Безусловно. Если вы хотя бы подозреваете, что в будущем может потребоваться повысить TQL инструмента, стоит ли просто выполнить дополнительную работу уже сегодня? Да, если есть избыточный бюджет и запас по графику. То есть, скорее всего, нет: почти ни один авиационный проект такой роскошью не обладает. Если существует разумная вероятность, что в будущем инструмент придется квалифицировать на более высокий TQL, дополнительные цели лучше отложить, за исключением независимости. Независимость относится к процессу верификации: рассмотрением, тестам и анализу. Если верификация не была выполнена независимо, то для более высокого TQL, где такая независимость требуется, ее придется выполнять заново. Рекомендация: выполняйте верификацию независимо, даже когда это формально не требуется. Независимому инженеру, возможно, понадобится немного больше времени на знакомство с техническими артефактами, но итоговая независимая верификация будет качественнее. Просто сделайте это.

Следующий рисунок обобщает требуемую строгость квалификации инструмента через количество целей, требуемых для каждого TQL:

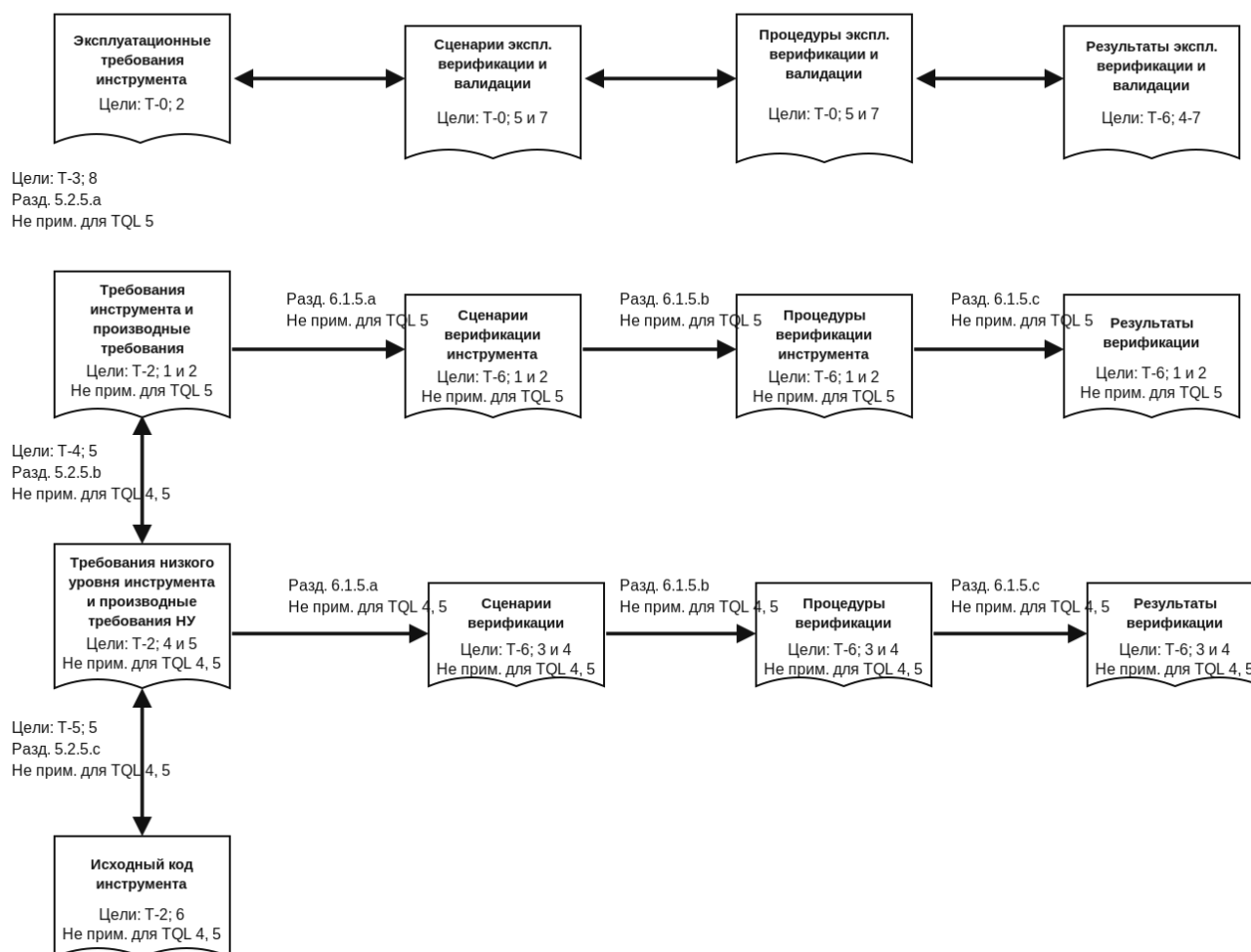
Ид.	Процесс	TQL-1	TQL-2	TQL-3	TQL-4	TQL-5
T-0	Эксплуатационные процессы инструмента	7	7	7	7	6
	- процесс планирования	1	1	1	1	1
	- процесс эксплуатационных требований к инструменту	1	1	1	1	1
	- процесс эксплуатационной интеграции инструмента	1	1	1	1	1
	- процесс эксплуатационной верификации и валидации инструмента	4	4	4	4	3
T-1	Процесс планирования инструмента	7	7	7	2	0
T-2	Процесс разработки инструмента	8	8	8	5	0
T-3	Верификация процессов требований к инструменту	9	9	9	8	0
T-4	Верификация процесса проектирования инструмента	11	11	9	1	0
T-5	Верификация процессов кодирования и интеграции инструмента	7	7	6	0	0
T-6	Тестирование процесса интеграции	4	4	4	2	0
T-7	Верификация тестирования инструмента	9	7	6	2	0
T-8	Процесс управления конфигурацией инструмента	5	5	5	5	2
T-9	Процесс обеспечения качества инструмента	5	5	5	2	2
T-10	Процесс взаимодействия по квалификации инструмента	4	4	4	4	4
Итого		76	74	70	38	14

Сводка целей квалификации инструмента по TQL

Какие связи трассируемости нужны для квалификации инструмента (TQ)?

Как и в других руководствах серии DO-XXX, учебная мантра AFuzion Incorporated “виновен, пока не доказана невиновность: докажи свою невиновность” вполне справедлива и для DO-330. На рисунке ниже обобщены различные доказуемые связи трассируемости, которые должны быть разработаны и оценены для разных требований квалификации инструмента в зависимости от TQL. Благодарность: Джону Линчу, AFuzion Inc, представителю FAA по инженерной экспертизе (Designated Engineering Representative, DER) и инженеру.

Трассировка требований и валидации/верификации по DO-330 Версия 4



Какая информация и какие данные требуются для квалификации инструмента (TQ)?

Формат и комплектование данных квалификации инструмента, которые нужно представить и сделать доступными для рассмотрения, зависят от типа и TQL квалифицируемого инструмента. Как и следовало ожидать, инструменты TQL 1 обычно требуют больше всего данных, а инструменты TQL 5 – меньше всего. Следующий рисунок 8 обобщает типовые данные квалификации инструмента по TQL и показывает, на

каком этапе цикла разработки они подготавливаются. Следует отметить, что сертифицирующие органы больше интересуются качеством данных квалификации инструмента, чем их упаковкой. Поэтому допускается достаточно широкая свобода: данные квалификации инструмента можно объединять в небольшом числе документов или даже включать в соответствующие документы разработки прикладного ПО. Может показаться разумным сократить количество документов, но есть нюанс: хорошие инструменты используют в других проектах или со временем настраивают, поэтому отдельные документы квалификации инструмента на самом деле упрощают повторное использование и повторную квалификацию. И помните: объем данных в каждом элементе на следующем рисунке зависит от TQL. Поэтому нужно внимательно прочитать все 138 страниц DO-330, чтобы включить требуемые данные и не собирать и не документировать то, что не требуется для менее строгих TQL.

Почему некоторые элементы данных на следующей схеме отмечены буквой “М” (Maybe, может требоваться)? При менее строгих TQL определенные данные либо не требуются, либо могут быть помещены в другие документы. Например, данные, которые обычно входят в план квалификации инструмента для инструмента TQL 5, можно просто включить в план сертификации ПО (Plan for Software Aspects of Certification, PSAC), тем самым устранив необходимость в отдельном плане квалификации инструмента. Просто. Один из вопросов, с которым придется столкнуться планировщику: выпускать ли отдельный план квалификации инструмента для каждого инструмента или один консолидированный план квалификации инструмента для всех инструментов. Некоторые группы выбирают второй вариант в текущем проекте, чтобы показать, как все инструменты интегрированы в процесс жизненного цикла.

Многие поставщики инструментов не предоставляют сертификат квалификации инструмента с полностью выполненным планом квалификации инструмента и полным комплектом квалификационных артефактов. Вместо этого они предоставляют пакет поддержки квалификации инструмента, который организация должна принять, адаптировать при необходимости, а затем выполнить, чтобы получить квалификационные артефакты. Крайне важно, чтобы организация, квалифицирующая инструмент, как можно раньше получила от поставщика пакет поддержки квалификации инструмента. Обычно такой пакет содержит: 1) шаблон плана квалификации инструмента; 2) типовые эксплуатационные требования к инструменту (Tool Operational Requirements, TOR); 3) план верификации инструмента; 4) процедуры квалификации инструмента и тестовые примеры; 5) файлы исходного кода или модели, необходимые для выполнения тестовых примеров; 6) проформу итогового отчета по инструменту. В некоторых случаях поставщик встраивает инструкции по адаптации в шаблоны документов, в других – предоставляет общий описательный документ. Важно рассмотреть и оценить качество инструкций по адаптации. Организации также должны оценить, какой объем поддержки по квалификации потребуется от поставщика для адаптации пакета поддержки квалификации инструмента, а иногда и для его выполнения.

Следует отметить, что поставщик инструмента обычно не предоставляет отчет об установке инструмента (Tool Installation Report, TIR), и организации важно хорошо разобраться именно с этим документом. В некоторых случаях предпочтительно выпускать единый консолидированный TIR для всех инструментов.

Элемент данных	Этап	TQL 1	TQL 2	TQL 3	TQL 4	TQL 5
Сведения о конкретном инструменте в плане сертификации ПО (PSAC)	Планирование и интегральные процессы	Да	Да	Да	Да	Да
План квалификации инструмента	Планирование и интегральные процессы	Да	Да	Да	Да	М
План разработки инструмента	Планирование и интегральные процессы	Да	Да	Да	Да	М
План верификации инструмента	Планирование и интегральные процессы	Да	Да	Да	Да	М
План управления конфигурацией инструмента	Планирование и интегральные процессы	Да	Да	Да	Да	М
План обеспечения качества инструмента	Планирование и интегральные процессы	Да	Да	Да	Да	М
Стандарты инструмента (требования, проект, код)	Планирование и интегральные процессы	Да	Да	Да	-	-
Указатель конфигурации инструмента	Планирование и интегральные процессы	Да	Да	Да	Да	М
Указатель конфигурации ЖЦ инструмента	Планирование и интегральные процессы	Да	Да	Да	Да	Да
Сообщения о проблемах инструмента	Планирование и интегральные процессы	Да	Да	Да	Да	Да
Записи управления конфигурацией инструмента	Планирование и интегральные процессы	Да	Да	Да	Да	Да
Записи обеспечения качества инструмента	Планирование и интегральные процессы	Да	Да	Да	Да	Да
Итоговый отчет по инструменту	Планирование и интегральные процессы	Да	Да	Да	Да	М
Сведения об инструменте в итоговом отчете разработки ПО	Планирование и интегральные процессы	Да	Да	Да	Да	Да
Требования к инструменту	Разработка инструмента	Да	Да	Да	Да	Да
Проект инструмента	Разработка инструмента	Да	Да	Да	М	-
Исходный код инструмента	Разработка инструмента	Да	Да	Да	Да	Да
Исполняемый код инструмента	Разработка инструмента	Да	Да	Да	Да	Да
Тесты верификации инструмента и результаты	Разработка инструмента	Да	Да	Да	Да	Да
Данные трассировки	Разработка инструмента	Да	Да	Да	Да	Да
Эксплуатационные требования к инструменту	Верификация и интеграция инструмента	Да	Да	Да	Да	Да
Отчет об установке инструмента	Верификация и интеграция инструмента	Да	Да	Да	Да	Да
Сценарии и процедуры эксплуатационной верификации и валидации инструмента	Верификация и интеграция инструмента	Да	Да	Да	Да	Да
Результаты эксплуатационной верификации и валидации инструмента	Верификация и интеграция инструмента	Да	Да	Да	Да	Да

Рисунок 8: Типовые данные квалификации заказного инструмента по TQL

Обозначения: Да – элемент данных обычно требуется; М – элемент данных может требоваться в зависимости от дополнительных критериев; пустая ячейка или дефис – элемент данных обычно не требуется.

Планирование, разработка и верификация инструмента

Как указано выше, для создания квалифицированных авионических инструментов есть три ключевых действия:

1. Планирование квалификации инструмента
2. Разработка инструмента
3. Верификация инструмента

Каждое из этих трех ключевых действий квалификации инструмента описано ниже.

Но сначала рассмотрим разные роли заинтересованных сторон инструмента: пользователя инструмента и разработчика инструмента, как показано на следующем рисунке:

Заинтересованные стороны инструмента: разработчик инструмента и пользователь инструмента

- **Разработчик инструмента** отвечает за разработку, верификацию, документирование и выпуск инструмента. Он разрабатывает инструмент для передачи продукта инструмента пользователю. Иногда могут быть определены дополнительные заинтересованные стороны, например интегратор или верификатор инструмента.
- **Интегратор или верификатор инструмента** удовлетворяет целям установки и эксплуатации инструмента, а также отвечает за квалификацию инструмента в процессе жизненного цикла ПО согласно руководящим указаниям.
- **Пользователь инструмента** отвечает за выбор, применение и квалификацию инструмента в процессе жизненного цикла ПО. Он также удовлетворяет целям установки и эксплуатации инструмента.
- **Заинтересованная сторона** включает пользователя и разработчика инструмента. Также может включать интегратора или верификатора. Заинтересованные стороны и их обязанности должны быть определены в планах: плане сертификации ПО (Plan for Software Aspects of Certification, PSAC) и плане квалификации инструмента.

1. Планирование квалификации инструмента (TQ) и данные

Неудивительно, что инженерия квалифицированных авионических инструментов очень похожа на инженерию авионики: спланировать, реализовать по планам, затем верифицировать. И все это при соблюдении интегральных процессов: сначала планирование, затем разработка, а управление конфигурацией, обеспечение качества и взаимодействие идут в фоновом режиме. У планирования квалификации инструмента много целей, включая следующие:

- Определить, а затем задать весь жизненный цикл инструмента и взаимосвязи между процессами жизненного цикла; обобщить это в плане сертификации ПО (PSAC) и/или в плане квалификации инструмента.
- Заранее идентифицировать среды разработки и верификации инструмента и связанные с ними детали.
- Установить применимые стандарты инструмента для требований, проекта и кода; обратите внимание, они могут быть очень похожи на аналогичные стандарты для авионического ПО или даже совпадать с ними.
- Идентифицировать все применимые цели DO-330 и определить, как именно будет достигаться каждая из них.

Выходом процесса квалификации инструмента будут применимые данные, включая следующие:

- **Сведения о конкретном инструменте в плане сертификации ПО (PSAC)**
- **План квалификации инструмента**
- **План разработки инструмента**
- **План верификации инструмента**
- **План управления конфигурацией инструмента**
- **План обеспечения качества инструмента**
- **Стандарты инструмента (требования, проект, код)**

2. Разработка инструмента и данные

После того как указанные выше данные планирования квалификации инструмента документированы и рассмотрены, начинается реализация инструмента или обратная разработка уже существующего инструмента в соответствии с этими планами. Применимые функциональные требования к инструменту, проект и код разрабатываются именно в таком порядке; при этом подтверждаются и проверяются критерии перехода и интеграции, включая трассируемость. Почему это важно и обязательно? Все просто: чтобы быть квалифицированным, инструмент должен пройти тщательную верификацию, как минимум тестирование функциональности инструмента относительно функциональности, заданной в его требованиях. Инструменты с более строгими TQL дополнительно проходят тестирование робастности и анализ структурного покрытия исходного кода инструмента. Но одно только тестирование само по себе никогда не обеспечивает высокое качество. Как и при строительстве небоскреба, испытаний на сейсмостойкость после завершения строительства недостаточно: такая стойкость должна быть заложена в архитектуру здания и учитываться при строительстве фундамента, стен и перекрытий. То же относится и к авионическим инструментам: требования к качеству должны учитываться на протяжении всего жизненного цикла разработки. Отсюда и цели DO-330 по планированию, процессам, стандартам требований, проекта и кодирования инструмента, интеграции, критериям перехода и трассируемости.

3. Верификация инструмента и данные

Как и сама сертифицируемая авиационная логика, квалифицированные инструменты, используемые для разработки или верификации этой логики, требуют определенного процесса верификации с конкретными целями. Верификация квалифицированных инструментов включает два последовательных процесса:

1. процесс верификации инструмента;

2. процесс эксплуатационной верификации и валидации инструмента.

Пункт 2 уникален для инструментов. Сертификация авиационного ПО (DO-178C) и аппаратуры (DO-254) не требует эксплуатационной верификации и валидации, поскольку эксплуатация такого полетного ПО и аппаратуры является частью требований системного уровня, а они верифицируются и валидируются именно на этом уровне. Однако, в отличие от авиационной аппаратуры и ПО, у инструментов нет требований системного уровня. Поэтому инструменты, квалифицируемые по DO-330, должны иметь дополнительный процесс, который подтверждает, что инструмент можно правильно эксплуатировать предполагаемым образом (верификация), а соответствующие эксплуатационные требования к инструменту корректны (валидация).

Требования к инструменту и эксплуатационные требования к инструменту. Да, квалифицированные инструменты требуют двух типов требований: требования к инструменту задают функциональность инструмента, а эксплуатационные требования к инструменту задают предполагаемое использование инструмента. Почему инструментам нужны два разных типа требований? Потому что инструментами обычно пользуются не те люди, которые эти инструменты разработали. Работа разработчиков инструмента, то есть сам инструмент, должна верифицироваться относительно предполагаемой функциональности, выраженной в требованиях к инструменту. Работа пользователя инструмента должна верифицироваться относительно предполагаемого использования, выраженного в эксплуатационных требованиях к инструменту. Между тем, что делает инструмент, и тем, как его эксплуатируют, есть связь, но требования к инструменту все равно отличаются от эксплуатационных требований к инструменту.

В заключение: при рассмотрении инструментов в процессе разработки никогда не вредно обратиться за помощью. ИЩИТЕ РУКОВОДЯЩИЕ УКАЗАНИЯ! Помните, что все инструменты нужно рассмотреть, отнести к соответствующей категории и, возможно, квалифицировать. Хорошая практика – указать в документах сертификационного планирования ВСЕ инструменты, которые предполагается использовать, и отметить, будет ли для них запрашиваться квалификация. Согласуйте категорию инструмента, необходимость его квалификации и методы квалификации как можно раньше.

Авиационная кибербезопасность DO-326A

Автор: Ахарон Давид

Рецензент:

г-н Вэнс Хилдерман, генеральный директор (Chief Executive Officer, CEO) и технический директор (Chief Technology Officer, CTO) AFuzion Inc.

Введение в авиационную кибербезопасность.

Худший враг: авиационные киберугрозы... или DO-326/ED-202?

Пассажир заходит на борт коммерческого авиалайнера с ноутбуком, взламывает его сеть и заставляет его лететь еще выше... смешно? К сожалению, это не начало шутки, а скорее начало потенциального кошмара.

Плохая новость? Нечто почти такое же, как описано выше, предположительно произошло в 2015 году... Хорошая новость? Этим человеком был “белый хакер” – один из “хороших”, кто пытается доказать необходимость усиления киберзащиты, поэтому самолет фактически оставался вне опасности.

Этому хакеру фактически закрыли доступ к будущим авиаперелетам, но Радиотехнической комиссии по аэронавтике (Radio Technical Commission for Aeronautics, RTCA) и Европейской организации по оборудованию для гражданской авиации (European Organisation for Civil Aviation Equipment, EUROCAE) даже не понадобился такой стимул: к тому времени они уже почти десять лет лихорадочно разрабатывали решение. Первый полный и пригодный к применению “комплект” документов DO-326/ED-202 наконец опубликовали в июне 2018 года, но еще до этого Федеральное управление гражданской авиации США (FAA) и Агентство Европейского союза по авиационной безопасности (EASA) сделали более ранние версии комплекта настолько обязательными, насколько это было практически возможно на тот момент. После публикации в середине 2018 года EASA уже признало комплект DO-326/ED-202 “приемлемым методом определения соответствия” (Acceptable Means of Compliance, AMC), то есть фактически обязательным стандартом; FAA сейчас рассматривает его так же и, как ожидается, вскоре должно последовать этому примеру.

Как мы пришли из той точки в нынешнюю? Хороший вопрос:

“Цифровой самолет” уже стал таким же привычным, как сам самолет, его крылья или двигатели. Современные самолеты вполне можно рассматривать как компьютеры с крыльями и силовой установкой.

На рубеже тысячелетий появился следующий цифровой авиационный феномен – “подключенный самолет”, где все связано... ну... со всем остальным. Сама тенденция не нова: цифровые радиосистемы, глобальная система позиционирования (Global Positioning System, GPS), адресно-отчетная система связи воздушного судна (Aircraft Communications Addressing and Reporting System, ACARS), автоматическое зависимое наблюдение-вещание, АЗН-В (Automatic Dependent Surveillance-Broadcast, ADS-B) – все это и многое другое уже десятилетиями входит в состав пассажирских самолетов. В итоге новые самолеты сегодня содержат тысячи процессоров, выполняющих как независимые, так и тесно связанные операции. Если старые самолеты имели сотни процессоров в “закрытой” системе, то современные архитектуры и сетевая связанность требуют большей открытости. Однако быстрый рост подключенности, ускоренный резким повышением производительности коммерческой аппаратуры и ПО, вместе с ранее неучтенными факторами сделал эту подключенность одновременно множителем преимуществ и источником угроз. Именно поэтому авиационная отрасль при координации RTCA (США) и EUROCAE (Европа) совместно разработала комплект нормативных документов DO-326/ED-202: нужно сохранить незаменимые преимущества подключенности, но не оставить гражданскую авиацию в целом и летную годность в частности без защиты от киберугроз, которые в итоге могут затронуть безопасность полетов. Этот комплект уже обязателен для ряда аспектов сертификации летной годности и быстро расширяет область применения, поэтому всем, кто работает в области летной годности или рядом с ней, лучше познакомиться с ним как можно раньше.

Комплект DO-326/ED-202 включает несколько документов общим объемом в несколько сотен страниц. Поэтому нужно понимать природу этой экосистемы, ее обязательные требования и возможные компромиссы. Прежде чем браться за эту экосистему кибербезопасности, необходимо разобраться с ее терминологией и процессами.

Есть несколько основных вопросов, которые необходимо рассмотреть до начала любого проекта по DO-326/ED-202:

- Что такое киберугроза / кибербезопасность?
- Как это связано с авиацией и воздушными судами?
- Какие руководства/стандарты по кибербезопасности существуют сегодня?
- В какой степени существующие стандарты применимы к авиации и воздушным судам?
- Что такое “DO-326/ED-202” и почему нельзя было просто применить “ARP-4754/DO-178”?
- Что такое “комплект DO-326/ED-202”?
- В какой степени “комплект DO-326/ED-202” является обязательным?

- Что представляют собой “руководства/рекомендации” “комплекта 326/202” (и чем они не являются...)?
- Что требуется, чтобы выполнить “руководства/рекомендации” “комплекта 326/202”?
- Как можно эффективно выполнить “руководства/рекомендации” “комплекта 326/202”?

Честные ответы на эти вопросы приведены ниже.

Что такое киберугроза/кибербезопасность?

Министерство внутренней безопасности США в своей оценке воздействия на конфиденциальность за 2010 год определяет “киберугрозу(ы)” как “...любую выявленную попытку без законных полномочий получить доступ к данным, приложению или федеральной системе, извлечь данные, изменить их либо нарушить их целостность, конфиденциальность, защищенность или доступность...”

К сожалению, каждый, кто живет в XXI веке, знаком с этой “цифровой угрозой”: компьютерные вирусы, черви, троянские программы и другие формы вредоносного ПО досаждают компьютерам со времен первого лабораторного вируса. Ниже – очень краткий исторический обзор эволюции этой цифровой угрозы:

Этап I – “Предыстория” (до всемирной паутины):

- [1971] “Creeping”: первый компьютерный вирус, разработанный Robert H. Thomas в BBN Technologies;
- [1982] “Elk Cloner”: первое вредоносное ПО, использовавшее вектор атаки; первоначально было создано для борьбы с пиратством в системах Apple II;
- [1988] червь Morris: заваливал компьютеры трафиком, мог заражать их многократно, вывел из строя около 6000 систем; Morris стал первым человеком, которого судили и признали виновным по американскому закону 1986 года “О компьютерном мошенничестве и злоупотреблениях” (Computer Fraud and Abuse Act);
- [1991] “Michaelangelo”, разновидность вируса “Stoned”: “просыпается” каждый год 6 марта (в день рождения художника), чтобы стереть первые 100 секторов локальных жестких дисков и гибких дисков.

Этап II – “Ранняя история” (эпоха всемирной паутины):

- [2000] червь “Love Letter”: электронное письмо с темой “ILOVEYOU” и вложенным текстовым файлом “LOVE-LETTER-FOR-YOU” заражает около 50 миллионов компьютеров за 10 дней, причиняя ущерб, оцениваемый в несколько миллиардов долларов США;

- [2007] троян “ZeuS”: “пакет”, содержащий различные “популярные” вредоносные программы, предназначенные для киберворов. Федеральное бюро расследований США арестовывает более 100 хакеров за банковское мошенничество в Восточной Европе после кражи 70 млн долларов с использованием “ZeuS”.

Этап III – “Переход в профессиональную лигу”:

- [2009] “Operation Aurora”: масштабные кибератаки – попытки проникнуть в репозитории исходного кода компаний – подрядчиков в сфере безопасности и обороны и вмешаться в них. Среди затронутых компаний, например: Google, Symantec и многие другие;
- [2010] вирус “Stuxnet” и связанные с ним кибератаки: первый случай, когда удалось эффективно шпионить за системами промышленного уровня и вмешиваться в их работу.

Этап IV – “Глобальная угроза”:

- [2013] программа-вымогатель “CryptoLocker”: кибератака распространялась через фишинговую схему по электронной почте, используя троян ZeuS. После загрузки полезная нагрузка программы-вымогателя шифровала файлы жесткого диска, “освобождая” их только после выплаты выкупа;
- [2016-2017] конкретные атаки, нацеленные на авиацию: как правило, направленные на аэропорты и/или авиакомпании; заметные жертвы – Вьетнам и Украина;
- [2017] атака программы-вымогателя “WannaCry”: всемирная синхронизированная атака, ущерб от которой оценивается примерно в 4 млрд долларов, включая инфраструктуру, в том числе участников гражданской авиации, таких как Boeing, LATAM Airlines Group и другие.

Эта короткая история – от наивного “экспериментального” вредоносного ПО до сегодняшней профессиональной, хорошо финансируемой глобальной киберпреступности – показывает, что под угрозой находится почти все: от частных компьютеров до стратегической инфраструктуры целых стран.

Так что, мы все умрем? Конечно, да, но, вероятно, от старости, а не от киберпреступности. Причина? Кибербезопасность. Но что такое кибербезопасность?

Кибербезопасность развивалась почти так же, как киберугрозы: постепенно, сначала как реакция на уже возникшие проблемы, а затем все более профессионально и совместно, по мере того как угрозы становились профессиональными и глобальными.

На домашнем уровне первыми появились простые меры защиты: “антивирус”, “антишпион”, затем общий класс “антивредоносных” средств, предназначенных для обнаружения и удаления вирусов, червей, троянов и вообще любого вредоносного ПО. Немного позже появились превентивные меры, известные как межсетевые экраны: они контролируют точки входа и блокируют атаки до их развития. Еще один тип контрмер,

хотя и не сугубо защитный, – устойчивость и восстановление: средства, которые позволяют системе работать хотя бы частично или в ухудшенном режиме даже во время атаки, а также резервное копирование и восстановление системы после повреждения. В современных, куда более сложных формах именно эти типы мер и сегодня составляют основу большинства технических средств защиты.

Но того, что достаточно для домашнего использования, едва ли хватит крупным организациям или объектам. Поэтому “технические меры защиты” со временем стали лишь одним из элементов более целостного подхода к кибербезопасности. В разных случаях он выглядит по-разному, но опирается на одни и те же базовые принципы: корпоративную стратегию, четкие роли и обязанности, явные кодексы и стандарты, архитектуру информационной безопасности сверху вниз и, в конечном счете, различные меры защиты: технические, организационные и другие.

Как кибербезопасность связана с авиацией и воздушными судами?

Пока индустрия информационных технологий (Information Technology, IT) могла отмахиваться от ранних любительских киберугроз как от “просто раздражающих, но не создающих реальной угрозы безопасности полетов”, нормой было самоуспокоение. Постепенно злоумышленники становились смелее, вредоносное ПО превращалось из статичных “зверьков” в сложные адаптирующиеся механизмы, способные сеять хаос, а накопленный ущерб рос экспоненциально, как уже описано выше.

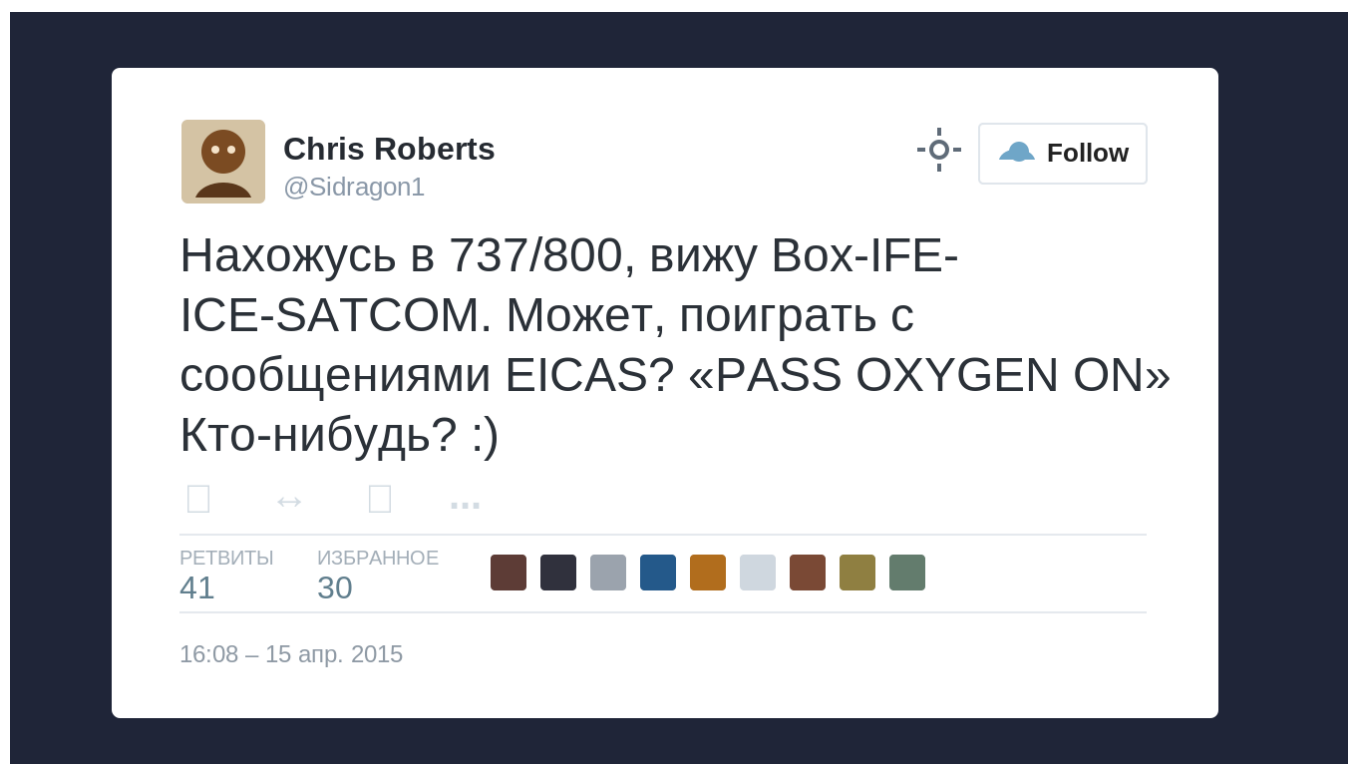
Эта тенденция быстро заставила сообщество IT-систем разрабатывать методы и средства нормальной кибербезопасности. А вот традиционная тяжелая промышленность, включая авиационную отрасль, двигалась в ту же сторону куда медленнее – вплоть до начала XXI века. Почему?

Более длительное самоуспокоение традиционных отраслей тяжелой промышленности перед лицом новых киберугроз объяснялось двумя основными соображениями: 1) системы информационных технологий и промышленные информационные системы имеют разную природу; 2) считалось, что столь сложные атаки на промышленную инфраструктуру способны выполнять только субъекты государственного уровня, а государства воспринимались как в разумной степени ответственные игроки, даже если речь шла о недружественных режимах. Это якобы создавало “естественное сдерживание” для катастрофических атак. В конечном итоге оба соображения оказались преувеличенными:

Промышленные информационные системы действительно отличаются от обычных систем информационных технологий. Они представлены несколькими распространенными разновидностями: операционные технологии (Operational Technology, OT), промышленные системы управления и автоматизации (Industrial Control & Automation Systems, ICAS), диспетчерское управление и сбор данных (Supervisory Control and Data Acquisition, SCADA), киберфизические системы (Cyber-Physical Systems, CPS) и другие. Типичные киберфизические системы раньше строились на проприетарных

технологиях и даже физически изолировались от стандартных IT-систем; такую изоляцию называли “воздушным зазором” (air gap). Оба свойства делали их практически недоступными для массовых хакеров. Однако примерно на рубеже тысячелетий готовые коммерческие изделия (Commercial Off-The-Shelf, COTS), включая аппаратуру и ПО, медленно, но верно проникли и в этот прочный сегмент, а “подключенный самолет” почти полностью ликвидировал “воздушный зазор”. По мере экспоненциального развития информационных и сетевых технологий и такого же резкого снижения их стоимости внезапно исчезло представление о барьере “государственного уровня”, который будто бы обеспечивал сдержанность: уже в первое десятилетие XXI века любой решительный хакер мог приобрести или даже разработать средства атаки, оказавшиеся слишком близко к авиационному миру. Даже предположение о благоразумии государств при крупных атаках внезапно перестало казаться надежным: мировой порядок дестабилизировался, а террористические организации, преступные группы и даже некоторые правительства перешли на “темную сторону”.

В результате во втором десятилетии XXI века появились кибератаки на аэропорты – главным образом во Вьетнаме и на Украине, – а также на инфраструктуру гражданской авиации, включая Boeing, LATAM и другие организации. Но самым драматичным событием, показавшим хрупкость предполагаемой защищенности, стало уже упомянутое в начале этой статьи:



К счастью, г-н Робертс – “белый хакер”, поэтому немедленного ущерба не было. Но предупреждение прозвучало громко и ясно и стало тем самым переломным моментом, который привел к сегодняшним новым руководящим материалам по кибербезопасности.

Какие руководства/стандарты по кибербезопасности существуют сегодня?

Руководящие материалы, а позднее и стандарты, объясняющие организациям, как правильно работать с кибербезопасностью, появились еще в 1970-х годах, когда воспринимаемая угроза была невелика. После нескольких десятилетий развития существуют буквально сотни различных типов стандартов, руководств и лучших практик по кибербезопасности, однако можно выделить три основные “семейства” гражданских стандартов плюс одно “семейство” военных стандартов как наиболее популярные:

1. **“Общие критерии” и “Общая методология” (Common Criteria, CC / Common Methodology, CM), также известные как ISO/IEC 15408:** первоначально объединение трех источников: американских критериев оценки доверенных компьютерных систем (Trusted Computer System Evaluation Criteria, TCSEC), также известных как стандарт Министерства обороны США (Department of Defense, DoD) 5200.28, корни которого уходят в 1970-е годы; канадских критериев оценки доверенных компьютерных продуктов (Canadian Trusted Computer Product Evaluation Criteria, CTCPEC) 1993 года; европейских критериев оценки безопасности информационных технологий (Information Technology Security Evaluation Criteria, ITSEC) 1990-х годов, принятых и в некоторых других странах. Этот общий комплект документов прежде всего стандартизирует терминологию и методы, используемые в кибербезопасности, и дает системам и организациям основу для подтверждения заявлений о кибербезопасности. Этот комплект документов не предлагает никаких конкретных, и даже общих, методов или решений.
2. **Комплект серии ISO/IEC 27000 (также известный как “ISO27K”):** это семейство стандартов системы управления информационной безопасностью, СУИБ (Information Security Management System, ISMS), представляющее собой систематический, всеобъемлющий набор из многих десятков конкретных стандартов, предназначенных для охвата всех аспектов кибербезопасности и широко принятых во всем мире как фактически обязательный стандарт. Его наиболее заметные документы – 27000 “Обзор и словарь” (Overview and vocabulary), 27001 “Требования” (Requirements), 27002 “Свод практик” (Code of practice), 27005 “Управление рисками” (Risk Management), и это лишь некоторые из них. ISO27K происходит от британского стандарта BS7799 1990-х годов, основанного на руководстве по политике информационной безопасности, разработанном Royal Dutch/Shell Group в конце 1980-х и начале 1990-х годов.

3. **Серия NIST SP-800:** с 1990-х годов Национальный институт стандартов и технологий США (NIST) публикует свою серию SP-800 с “... информацией, представляющей интерес для сообщества компьютерной безопасности...”, которая “... включает руководства, рекомендации, технические спецификации и ежегодные отчеты о деятельности NIST в области кибербезопасности...” и стала незаменимым источником знаний, послужив основой для большинства других стандартов по кибербезопасности, включая другие известные “семейства”. Документы этой серии широко используются как фактические стандарты, особенно для аспектов, которые еще недостаточно хорошо охвачены другими стандартами. Помимо серии SP-800, NIST опубликовал несколько крупных программных документов, которые также широко используются во всем мире, например NIST “Структура кибербезопасности” (Cybersecurity Framework), Федеральные стандарты обработки информации (Federal Information Processing Standards, FIPS) 200: “Минимальные требования безопасности для федеральной информации и информационных систем” (Minimum Security Requirements for Federal Information and Information Systems), сделавший SP-800 фактическими стандартами, и некоторые другие.

4. **Серия DoD 8500:** набор директив и инструкций Министерства обороны США, направленных на “гарантию выполнения задач” (Mission Assurance), а именно на “гарантию информационной безопасности” (Information Assurance), в которой кибербезопасность является инструментом обеспечения выполнения военных задач.

Все эти и многие другие документы используются по всему миру как стандарты и/или лучшие практики для различных целей кибербезопасности.

В какой степени существующие стандарты применимы к авиации и воздушным судам?

Хотя любое из основных “семейств” стандартов по кибербезопасности могло и действительно может служить полезной справочной базой для авиационных стандартов кибербезопасности, в 2005 году, когда стандартизация авиационной кибербезопасности началась всерьез, у всех этих семейств были существенные пробелы там, где на кону стояла безопасность полетов.

Все эти проблемы были связаны с природой главных “активов”, которые должны защищать такие стандарты кибербезопасности: пассажиров на борту коммерческих самолетов, а значит, сами самолеты и их критически важные системы:

1) Информационные технологии против операционных технологий: почти все системы воздушного судна, а также многие авиационные системы управления и наземной поддержки являются киберфизическими системами и операционными технологиями, а не чистыми ИТ-системами. Одно это различие почти исключает применение любого из основных семейств стандартов кибербезопасности, описанных выше.

Кибербезопасность операционных технологий и киберфизических систем, как уже говорилось, стартовала поздно. Однако в 2002 году Международное общество автоматизации (International Society for Automation, ISA) и Американский национальный институт стандартов (American National Standards Institute, ANSI) запустили комитет по стандартам ANSI/ISA-99, который начал работу над комплектом стандартов “Защита промышленных систем автоматизации и управления” (Industrial Automation and Control Systems Security). Основой стал документ NIST SP-800-82 “Руководство по защите промышленных систем управления” (Guide to Industrial Control Systems (ICS) Security), единственный на тот момент жизнеспособный стандарт по кибербезопасности операционных технологий. ANSI/ISA-99 несколько раз менял название, и сейчас этот комплект стандартов известен как ISA/IEC 62443.

Отличное решение? Едва ли, и по двум основным причинам. Во-первых, авиация действительно основана на операционных технологиях и киберфизических системах, но также включает обычные ИТ-системы и множество дополнительных соображений, которые невозможно учесть в универсальном комплекте стандартов по принципу “один размер подходит всем”. Во-вторых, и это еще серьезнее, к тому времени, когда авиационный мир был готов к выработке стандартов кибербезопасности, примерно к 2005 году, ANSI/ISA-99 еще не был готов. Более того, даже к 2020 году были опубликованы только 8 из 13 запланированных документов комплекта, и еще одна часть должна была выйти в ближайшее время.

2) Военные против гражданских: стандарты военного типа, такие как комплект DoD 8500, не подходили по той же причине, по которой военные стандарты безопасности полетов не были приняты в гражданской авиации: военные стандарты обычно ориентированы на выполнение задачи, то есть имеют функциональный характер, а гражданские стандарты ориентированы на общественную безопасность с разумным запасом по характеристикам. Даже само название DoD 8500 – “гарантия выполнения задач” (Mission Assurance), так что этот вариант всерьез не рассматривался.

3) Другие отрасли: на основе общих стандартов кибербезопасности было разработано немало отраслевых стандартов. Некоторые отрасли достаточно близки к авиации, чтобы их можно было всерьез рассматривать как исходную базу для авиационных требований, но в 2005 году такого выбора еще не было. Ближайший такой “родственник” – документ SAE J3061 “Руководство по кибербезопасности для киберфизических автомобильных систем” (Cybersecurity Guidebook for Cyber-Physical Vehicle Systems), опубликованное в 2016 году и в 2020 году проходившее совместный пересмотр ISO и SAE для выпуска нового стандарта: ISO/SAE 21434 “Дорожные транспортные средства. Инженерия кибербезопасности” (Road Vehicles – Cybersecurity Engineering).

4) Существующие авиационные стандарты: в 2005 году практически не существовало “действующих авиационных стандартов по кибербезопасности”, по крайней мере общего назначения, и особенно стандартов фазы разработки. У Международной организации гражданской авиации (International Civil Aviation Organization, ICAO) было свое

руководство по авиационной безопасности, “Приложение 17” (Annex 17), но киберугрозы были добавлены в него лишь в 2011 году в виде двух абзацев верхнего уровня. Ассоциация воздушного транспорта Америки (Air Transport Association of America, ATA), ныне Airlines for America (A4A), выпустила свой документ “Спецификация 42. Отраслевые авиационные стандарты цифровой информационной безопасности” (Spec 42: Aviation Industry Standards for Digital Information Security), содержащий около 400 страниц технических руководящих указаний по шифрованию для авиации, но только в 2008 году.

ARINC (Aeronautical Radio, Incorporated) продвинулась несколько дальше и в 2005 году выпустила документ “Спецификация 664, часть 5. Сеть передачи данных воздушного судна” (Specification 664 Part 5: Aircraft Data Network), который был и остается весьма полезным документом по вопросам защищенных сетей воздушного судна. В том же году ARINC опубликовала технический отчет 811 “Концепция эксплуатации и рамочная структура процессов информационной безопасности коммерческого самолета” (Technical Report 811, Commercial Aircraft Information Security Concepts of Operation and Process Framework). Он действительно был ориентирован скорее на авиакомпанию и эксплуатацию, чем на разработку, но содержал судьбоносную рекомендацию:

”... Поощрять гармонизацию существующих авиационных практик гарантии, включая RTCA DO-160D, а также практик/стандартов гарантии разработки (например, DO-178B), с практиками/стандартами информационной безопасности (например, NIST FIPS-140 и Common Criteria).

Примечание: необходимо свести эти две области вместе и предоставить авиакомпаниям и регуляторам общую основу для оценки решений по информационной безопасности воздушных судов, например при оценке готовых коммерческих (COTS) решений информационной безопасности...”

Что такое “DO-326/ED-202” и почему нельзя было просто применить “ARP4754/DO-178”?

Как уже отмечалось, когда европейская и американская авиационная промышленность в координации с EASA и FAA примерно в 2005 году приступила к выработке стандартов, ни один из существовавших тогда стандартов кибербезопасности не давал готового решения. Поэтому в 2006 году EUROCAE создала рабочую группу 72 (Working Group 72, WG-72), а в 2007 году RTCA создала специальный комитет 216 (Special Committee 216, SC-216); обе структуры получили название “Информационная безопасность авиационных систем” (Aeronautical Systems Security). Так начался процесс, результатом которого стали DO-326/ED-202 и сопутствующие документы экосистемы.

Первые документы, ED-202 в Европе и DO-326 в США, оба под названием “Спецификация процесса информационной безопасности летной годности” (Airworthiness Security Process Specification), были опубликованы в 2010 году. Изначально DO-326/ED-202 должны были служить руководством “все в одном” по информационной

безопасности на фазе разработки самолета – от замысла до сертификации и ввода в эксплуатацию. В исходном тексте документов прямо говорилось, что они являются “... первым из серии документов по информационной безопасности авиационных систем, которые совместно будут охватывать информационную безопасность в рамках общей информационной безопасности авиационных информационных систем (Aeronautical Information System Security, AISS) бортовых систем вместе со связанными наземными системами и средой...”, но эта фаза была еще впереди.

DO-326/ED-202 в значительной степени основывались на уже опубликованном ISO/IEC 27005 из семейства ISO27K, а также на фактическом отраслевом стандарте SAE ARP 4754 (Aerospace Recommended Practice, ARP) “Соображения сертификации для высокоинтегрированных или сложных авиационных систем” (Certification Considerations for Highly-Integrated or Complex Aircraft Systems) и достаточно плавно выстраивали с ними полезную преемственность.

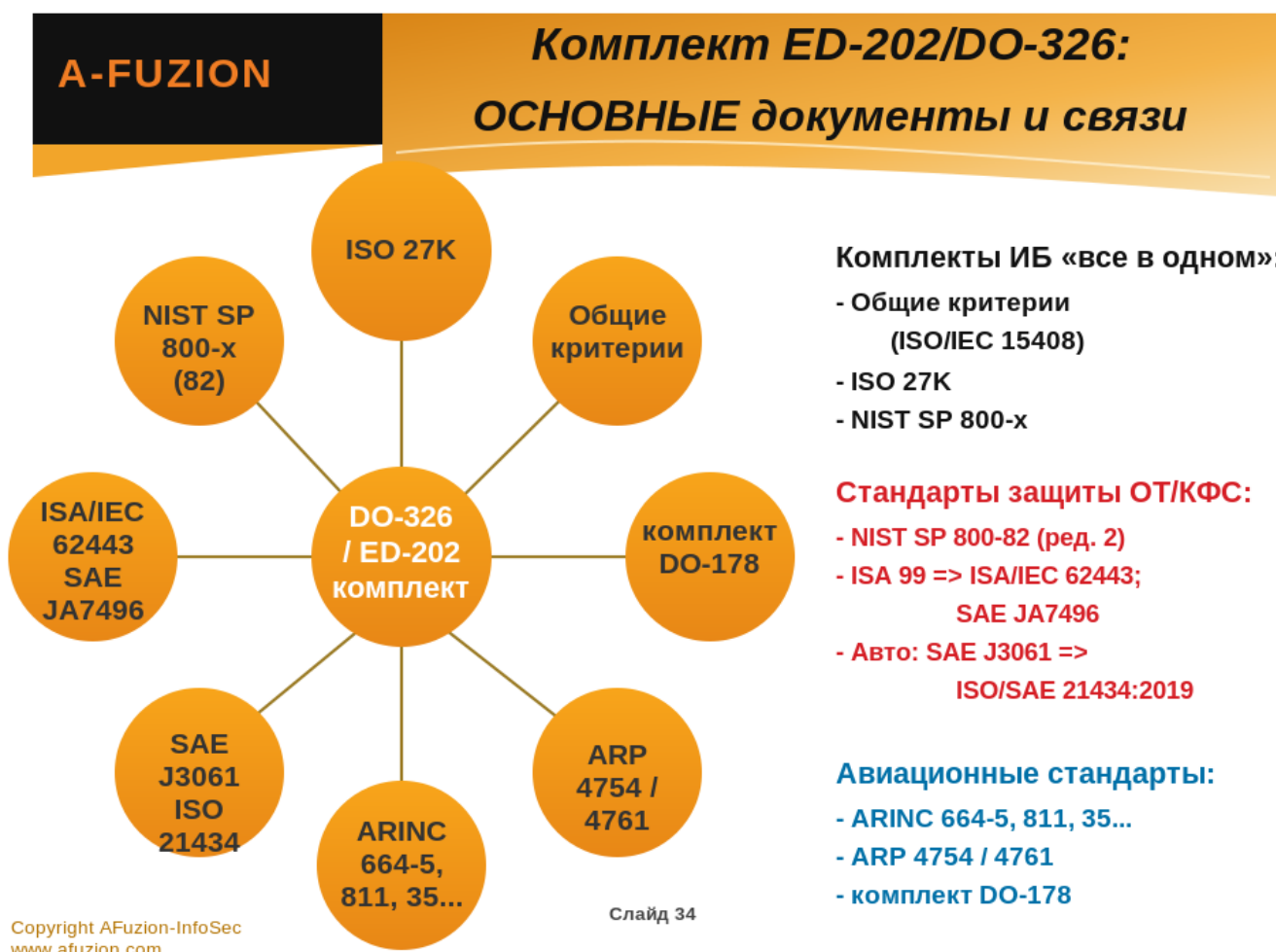
Однако любой, кто хотя бы немного знаком с ARP4754 и/или RTCA DO-178 “Соображения по ПО при сертификации бортовых систем и оборудования” (Software Considerations in Airborne Systems and Equipment Certification), вправе спросить, зачем вообще потребовались все эти усилия: почему информационную безопасность нельзя было просто включить в ARP4754 и/или DO-178, особенно если оба документа тогда проходили крупные пересмотры?

Простая практическая причина изложена в сопутствующем документе DO-326/ED-202, опубликованном несколькими годами позже: “... информационная безопасность летной годности является самостоятельной дисциплиной, требующей уникальной экспертизы, а также собственных методов анализа и соображений гарантии...”. Иными словами, она отличается и от безопасности полетов, и от других аспектов информационной безопасности. Еще одно соображение состояло в том, что информационная безопасность вовсе не сводится к ПО: она затрагивает множество других аспектов авиации. Поэтому ARP 4754 и DO-178 нужно было скоординировать с ней, но сохранить “чистыми” от вопросов кибербезопасности.

Последним препятствием для интеграции информационной безопасности в ARP 4754 и/или DO-178 был тот факт, что их “источник легитимности”, документ FAA/EASA AC/AMC 25.1309 “Проектирование и анализ систем” (Advisory Circular / Acceptable Means of Compliance, AC/AMC; System Design and Analysis), прямо исключает такую возможность уже самим определением термина “событие”, охватываемого документом: “Событие: происшествие, источник которого находится вне воздушного судна, например атмосферные условия (например, порывы ветра, изменения температуры, обледенение и удары молнии), состояние взлетно-посадочной полосы, состояние служб связи, навигации и наблюдения, столкновение с птицей, пожары в салоне и багажных отсеках... Термин не предназначен для охвата саботажа...”. Понимая, что любое изменение формальных документов FAA/EASA может задержать процесс примерно на десятилетие, и реагируя на

необходимость немедленно закрыть нормативный пробел в информационной безопасности, WG-72 и SC-216 получили поддержку в разработке отдельных документов, тесно скоординированных с ARP 4754 и DO-178. Именно это они и сделали.

И хотя комплект DO-326/ED-202 все еще развивается, SAE уже разрабатывает соответствующую экосистему – сейчас это JA7496 и JA6678. Эти документы должны дать технический контекст как комплекту DO-326/ED-202, так и автомобильным стандартам кибербезопасности.



Что такое “комплект DO-326/ED-202”?

Как уже отмечалось, исходные документы DO-326/ED-202 первоначально задумывались как первые документы серии по информационной безопасности авиационных систем. И действительно, после первой публикации 2010 года WG-72/SC-216 продолжили разработку этой серии.

Хотя WG-72 были (и остаются) тесно скоординированы, а многие участники состоят в обеих группах, между ними все же имеются некоторые различия, главным образом в базовой философии: в то время как SC-216 (США) придерживается прямолинейного подхода, ориентированного на быстрые результаты, и почти исключительно сосредоточена на информационной безопасности самолетов (“... руководящие материалы по защищенности самолетных систем...”), WG-72 (Европа) приняла более целостный

подход, охватывающий больше аспектов авиационной информационной безопасности (“WG-72 будет применять целостный подход, рассматривая темы, связанные с информационной безопасностью, на протяжении всего жизненного цикла изделий...”). Целостный подход WG-72 был легко узнаваем с самого начала по выделению всего диапазона номеров документов ED-20х для развивающейся серии, тогда как SC-216 просто использовала последовательную сквозную нумерацию всей серии DO.

Первые шаги после публикации DO-326 и ED-202 все еще были тесно скоординированы:

1. “Очистить” DO-326/ED-202, чтобы сделать его “базовым” документом, который охватывал бы только “ЧТО” сертификационного процесса. Пересмотренная версия DO-326A/ED-202A была опубликована в 2014 году.
2. Опубликовать новый документ DO-355/ED-204 “Руководящие указания по информационной безопасности для поддержания летной годности” (Information Security Guidance for Continuing Airworthiness), чтобы охватить фазу после производства, то есть эксплуатацию; также в 2014 году. Новый DO-355/ED-204 включил части, выделенные из DO-326/ED-202.
3. Опубликовать новый сопутствующий документ, рассматривающий “КАК” сертификационного процесса. Новая пара DO-356/ED-203 “Методы и соображения по информационной безопасности летной годности” (Airworthiness Security Methods and Considerations) задумывалась как “аналог DO-178 для информационной безопасности” к “аналогу ARP4754 для информационной безопасности” в лице DO-326. В конечном итоге эта эквивалентность 4754/178 не была выдержана, и оба документа включают части, идущие параллельно как ARP4754, так и DO-178.

Однако на этом третьем этапе различная природа WG-72 и SC-216 начала сказываться, поскольку итоговые DO-356 (опубликован в 2014 году) и ED-203 (опубликован в 2015 году) приняли разные подходы и не были технически идентичны, как другие элементы “серии”. Кроме того, WG-72, реализуя свой целостный подход, в 2015 году опубликовала собственный документ без эквивалента SC-216: ED-201 “Руководство по рамочной структуре информационной безопасности авиационных информационных систем” (Aeronautical Information System Security Framework Guidance), задуманный как стратегический документ верхнего уровня, обеспечивающий “общую картину” и поддерживаемый двумя фоновыми отчетами: ER013 “Глоссарий по информационной безопасности авиационных информационных систем” (Aeronautical Information System Security Glossary) в 2015 году и ER17 “Сводка международной карты деятельности по авиационной информационной безопасности” (International Aeronautical Information Security Activity Mapping Summary) в 2018 году.

На этом этапе комитет FAA по нормотворчеству, координирующий действия с EASA, Национальным агентством гражданской авиации Бразилии (ANAC) и Министерством транспорта Канады (TCCA), вмешался и среди множества других рассматриваемых

вопросов вновь сблизил комитеты EUROCAE и RTCA, что привело к пересмотру DO-356 и ED-203, опубликованных в 2018 году как DO-356A/ED-203A, с унифицированным техническим содержанием. Однако ED-201 не получил эквивалента в серии DO, поскольку не считался “рабочим документом”, а был признан “ориентирующим документом”.

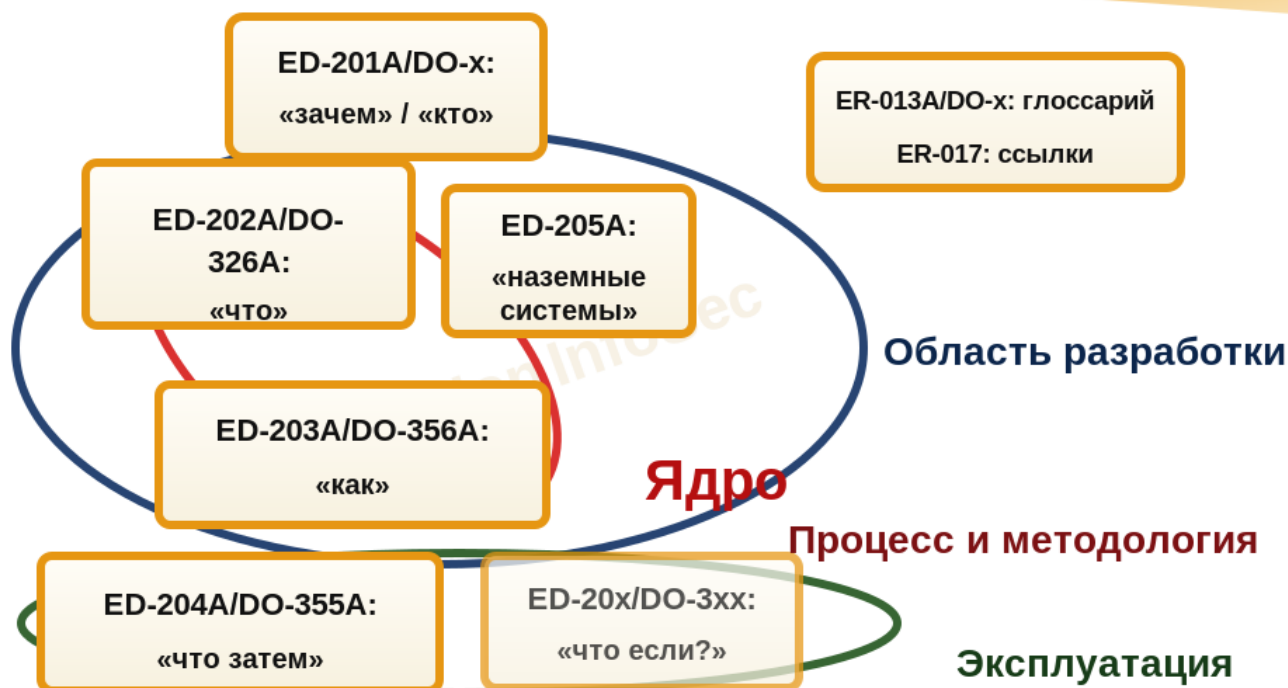
Тем временем WG-72 в 2019 году опубликовала еще один собственный документ, ED-205 “Стандарт процесса для сертификации/декларирования информационной безопасности наземных систем организации воздушного движения и аэронавигационного обслуживания” (Process Standard for Security Certification/Declaration of Air Traffic Management/Air Navigation Services, ATM/ANS, Ground Systems), опять же без немедленного эквивалента SC-216, однако по состоянию на декабрь 2020 года RTCA решила присоединиться к этой работе со своей собственной версией.

Таким образом, по состоянию на 2021 год комплект DO-326/ED-202 включает:

1. “Базовые” руководящие материалы и соображения: DO-326A/ED-202A и DO-356A/ED-203A;
2. “Эксплуатационные” руководящие материалы: DO-355/ED-204, основанные на “базовых” руководящих материалах;
3. Документы “верхнего уровня”: ED-201, ER-013, ER-017, служащие “философскими руководящими принципами”;
4. Стандарт для наземных систем ED-205, в основном предназначенный для европейского применения.

В течение следующих нескольких лет WG-72 / специальный комитет 210 (Special Committee 210, SC-210) планируют выпустить:

1. Новый сопутствующий документ DO/ED “Управление событиями информационной безопасности” (Information Security Event Management) для поддержки DO-355/ED-204;
2. Редакцию А для DO-355/ED-204, которая уже опубликована, но все еще должна быть включена в формальное регулирование в Европе и США;
3. Редакцию А для ED-201 и новый документ DO, эквивалентный ED-201A;
4. Редакцию А для ED-205 и новый документ DO, эквивалентный ED-205A;
5. Редакцию А для ER-013 и новый документ DO, эквивалентный ER-13A.



В какой степени “комплект DO-326/ED-202” является обязательным?

В Европе 1 июля 2020 года EASA опубликовало свое “решение ED 2020/006/R”, приняв с незначительными изменениями выпущенное EASA в феврале 2019 года уведомление о предлагаемой поправке (Notice of Proposed Amendment, NPA) 2019-01 “Кибербезопасность воздушных судов” (Aircraft cybersecurity), **вступающее в силу с 1 января 2021 года**.

Решение ED 2020/006/R вносит изменения в CS-25, CS-27, CS-29, CS-APU, CS-E, CS-ETSO, CS-P, соответствующие приемлемые методы определения соответствия (AMC) и руководящие материалы (Guidance Material, GM), а также в AMC-20, AMC/GM к CS-23 и AMC/GM к Part 21.

Поправка к AMC-20, добавляющая AMC 20-42 “Оценка рисков информационной безопасности летной годности” (Airworthiness information security risk assessment), практически делает базовые документы комплекта ED-202 (и их эквивалент, американский комплект DO-326) **ЕДИНСТВЕННЫМ** методом определения соответствия для аспектов кибербезопасности в сертификации летной годности:

Еще одна задача EASA по нормотворчеству RMT.0720 (Rule Making Task, RMT) “Кибербезопасность авиационных систем” (Aviation System Cybersecurity), уже выпустила предлагаемые поправки к существующему регулированию, NPA 2019-07, для всего

авиационного сектора, чтобы дополнить упомянутое выше регулирование летной годности; ожидается, что они будут приняты с некоторыми изменениями в период 2021-22 годов и сделают ED-201 и ED-205 также “приемлемыми методами определения соответствия”.

В США ранее упомянутый комитет FAA по нормотворчеству, Консультативный комитет по разработке авиационных правил (FAA Aviation Rulemaking Advisory Committee, ARAC), и рабочая группа по информационной безопасности и защите самолетных систем (Aircraft System Information Security / Protection Working Group, ASISP WG), которая в 2015-2016 годах рассматривала все регулирование информационной безопасности, значительно ускорили процесс установления формального обязательного регулирования, и их весьма ясная рекомендация по этому вопросу состояла в том, чтобы Федеральное управление гражданской авиации США “... рассматривало стандарты RTCA DO-326, DO-356 и DO-355 и стандарты EUROCAE ED-201, ED-202, ED-203, ED-204 как приемлемые руководящие материалы для соблюдения правила по информационной безопасности 25.13xx для крупных транспортных самолетов при новой сертификации типа, новых значительных крупных изменениях или когда заявитель добровольно решает их использовать...”, что на языке регулятора переводится как “Федеральное управление гражданской авиации США: сделать это обязательным, сейчас”. Поскольку EASA, Национальное агентство гражданской авиации Бразилии (ANAC) и Министерство транспорта Канады (TCCA) активно сотрудничали с этим комитетом и этой рабочей группой, ясный смысл этого таков: комплект DO-326/ED-202 становится обязательным, и очень скоро – во всем мире. Эта рекомендация хорошо согласуется с рабочей группой EASA RMT.0648, разрабатывающей “Сертификационные спецификации для проектирования изделий” (Certification Specifications for product design), и рабочей группой RMT.0720, разрабатывающей “Правила управления рисками в организациях и у поставщиков услуг” (Rules for risk management within organizations and service providers) – оба этих “горизонтальных” руководящих материала обозначают комплект DO-326/ED-202 и были запланированы к публикации в 2020 году.

Другими критически важными рекомендациями этой рабочей группы были ретроактивное проведение “кабинетного обзора” (Table-top Review) для **существующих** бортовых технических стандартных приказов (Technical Standard Orders, TSO) и стандартов на оборудование связи, навигации, наблюдения и организации воздушного движения (Communication, Navigation, Surveillance / Air Traffic Management, CNS/ATM), а также выработка руководящих указаний по использованию готового коммерческого ПО и **ранее сертифицированных** изделий.

Поскольку этот комитет и эта рабочая группа также рекомендовали адаптировать комплект DO-326/ED-202 ко всем категориям самолетов, двигателей и воздушных винтов, вместе с весьма недвусмысленными формулировками для новых правил и нормативных требований, совершенно ясный ответ таков: всем участникам авиационного бизнеса следует быть готовыми к тому, что DO-326/ED-202 станет полностью обязательным не более чем в течение десятилетия, а возможно – значительно раньше.

Некоторые аспекты комплекта DO-326/ED-202 уже являются обязательными:

Заявление политики FAA PS-AIR-21.16-02, редакция 2: “Установление специальных условий для защиты информационной безопасности самолетных систем” (Establishment of Special Conditions for Aircraft Systems Information Security Protection) 2017 года гласит, что Федеральное управление гражданской авиации США “... будет выпускать специальные условия для заявок на первоначальный сертификат типа (Type Certificate, TC), дополнительный сертификат типа (Supplemental Type Certificate, STC), измененный сертификат типа или измененный дополнительный сертификат типа для систем воздушного судна, подключающихся к недоверенным службам (например, негосударственным) и сетям...” для всех категорий воздушных судов: самолетов транспортной категории по 14 CFR Part 25, самолетов пригородной категории по 14 CFR Part 23, винтокрылых летательных аппаратов нормальной категории с несколькими двигателями по 14 CFR Part 27 и винтокрылых летательных аппаратов транспортной категории по 14 CFR Part 29. Поскольку указанная редакция 2 является продуктом этого комитета и этой рабочей группы, а комплект DO-326/ED-202 был рекомендован как приемлемый метод определения соответствия, смысл этой политики состоит в том, что фактически комплект DO-326/ED-202 теперь является нормой для данного типа сертификации для всех категорий воздушных судов.

Консультативный циркуляр FAA AC 119-1: “Разрешение по летной годности и эксплуатации программы защиты сетей воздушного судна” (Airworthiness & Operational Authorization of Aircraft Network Security Program, ANSP) 2015 года использует “эксплуатационную” часть комплекта: DO-355, который опирается на DO-326 и DO-356.

Консультативный циркуляр FAA AC 20-140C: “Руководящие указания по одобрению конструкции систем передачи данных воздушного судна, поддерживающих обслуживание воздушного движения (ОВД)” (Guidelines for Design Approval of Aircraft Data Link Communication Systems Supporting Air Traffic Services, ATS) 2015 года использует “базовый” и “эксплуатационный” полный комплект: DO-326, DO-356 и DO-355.

Консультативный циркуляр AC 120-76D: “Разрешение на использование электронных полетных сумок” (Authorization for Use of Electronic Flight Bags) 2017 года включает новый раздел “Процедуры информационной безопасности” (Security Procedures), который использует циркуляр AC 20-140C, а тот опирается на весь комплект DO-326.

Обратите внимание: новые обязательные требования уже появились, так что игнорировать комплект DO-326/ED-202 приходится на свой страх и риск...

Что представляют собой руководящие материалы и рекомендации комплекта DO-326/ED-202? (...и чем они не являются...)

Руководящие материалы и рекомендации комплекта DO-326/ED-202 распределены по всем входящим в него документам. Но для целей летной годности его удобно рассматривать как комплект из двух частей: (1) DO-326/ED-202 и DO-356/ED-203 – базовые документы, относящиеся к фазе разработки; (2) DO-355/ED-204 – документы по поддержанию летной годности в эксплуатации. Чтобы соблюдать комплект надлежащим образом, заявителю лучше воспринимать его именно как комплект, то есть как единый корпус текста для разработки, эксплуатации или обеих областей, а не как разрозненную коллекцию документов.

Ни одна часть комплекта и ни один конкретный документ не предписывают конкретные меры, техники или методы защиты, которые обязательно нужно внедрить. Поэтому комплект DO-326/ED-202 ни в коем случае не следует воспринимать как “кулинарную книгу”. Как следует из названий документов, это руководящие материалы – от стратегии верхнего уровня до подробной тактики. Во многих случаях они потребуют применения дополнительных стандартов информационной безопасности; некоторые из них, например ISO 27K, прямо рекомендованы самим комплектом.

Что касается руководящих материалов/рекомендаций, включенных в комплект, их можно примерно описать следующим образом:

1) “Процесс информационной безопасности летной годности” (Airworthiness Security Process, AWSP), подробно изложенный главным образом в DO-326A/ED-202A. Он описывает основные шаги, мероприятия и цели сертификации летной годности в части информационной безопасности:

- a. Процесс информационной безопасности летной годности включает 7 шагов: план по аспектам информационной безопасности при сертификации (Plan for Security Aspects of Certification), определение области информационной безопасности (Security Scope Definition), оценка рисков информационной безопасности (Security Risk Assessment), определение приемлемости риска (Risk Acceptability Determination), разработка мер защиты (Security Development), гарантия эффективности защиты (Security Effectiveness Assurance) и представление свидетельств (Communication of evidences).
- b. Эти 7 шагов развернуты в 14 мероприятий: план по аспектам информационной безопасности при сертификации (Plan for Security Aspects of Certification, PSecAC), краткое изложение этого плана (PSecAC Summary), определение области информационной безопасности воздушного судна (Aircraft Security Scope Definition, ASSD), предварительная оценка рисков информационной безопасности воздушного судна (Preliminary Aircraft Security Risk Assessment, PASRA), оценка рисков информационной безопасности воздушного судна (Aircraft Security Risk Assessment, ASRA), определение области информационной безопасности системы (System Security Scope Definition, SSSD), предварительная оценка рисков информационной

безопасности системы (Preliminary System Security Risk Assessment, PSSRA), оценка рисков информационной безопасности системы (System Security Risk Assessment, SSRA), архитектура и меры защиты воздушного судна (Aircraft Security Architecture and Measures, ASAM), руководящие указания по информационной безопасности для эксплуатанта воздушного судна (Aircraft Security Operator Guidance, ASOG), верификация информационной безопасности воздушного судна (Aircraft Security Verification, ASV), архитектура и меры защиты системы (System Security Architecture and Measures, SSAM), руководящие указания по информационной безопасности для системного интегратора (System Security Integrator Guidance, SSIG) и верификация информационной безопасности системы (System Security Verification, SSV).

- с. Эти 14 мероприятий в совокупности включают 62 цели, которые должны быть достигнуты.
- d. Подробные приемлемые методы определения соответствия приводятся в основном в DO-356A/ED-203A.

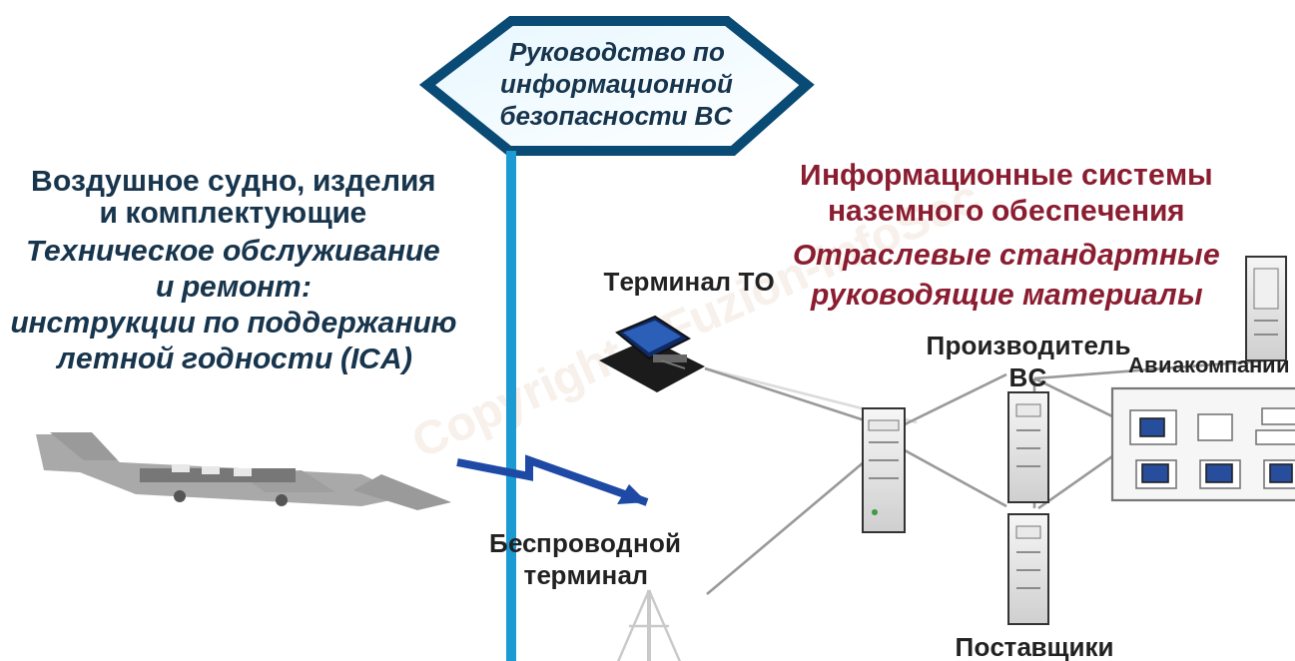


Copyright AFuzion-InfoSec
www.afuzion.com

2) “Руководство по поддержанию летной годности”, в основном задаваемое DO-355/ED-204, но также поддерживаемое базовыми документами комплекта. Это руководство охватывает одиннадцать аспектов: обращение с бортовым ПО, обращение с компонентами воздушного судна, точки доступа к сети воздушного судна, наземное вспомогательное оборудование (Ground Support Equipment, GSE), информационные системы наземного

обеспечения (Ground Support Information Systems, GSIS), цифровые сертификаты, управление инцидентами информационной безопасности воздушного судна, программа информационной безопасности воздушного судна эксплуатанта, оценка рисков организации эксплуатанта, роли и обязанности персонала эксплуатанта, обучение персонала эксплуатанта.

В совокупности процесс информационной безопасности летной годности и руководство по поддержанию летной годности образуют единую интегрированную сущность, представляющую собой полный комплект приемлемых методов определения соответствия по информационной безопасности для достижения и поддержания летной годности на протяжении всего жизненного цикла воздушного судна.



Что требуется, чтобы выполнить руководящие материалы и рекомендации комплекта DO-326/ED-202?

Хотя DO-326A/ED-202A задает формальные требования высокого уровня к процессу информационной безопасности летной годности, DO-355/ED-204 более подробно определяет, что требуется для сохранения летной годности. Но и здесь акцент сделан на надлежащем процессе и следовании более детальным общим стандартам информационной безопасности, с четкими ролями для держателя одобрения проекта (Design Approval Holder, DAH), обычно разработчика оборудования, и эксплуатанта оборудования.

Что касается самого процесса сертификации летной годности, здесь в основном помогает DO-356A/ED-203A. Этот 370-страничный документ подробно описывает особенности информационной безопасности в трех ключевых подпроцессах сертификации авионики: планировании (включая область информационной безопасности), разработке и интегральном процессе (включая оценку рисков и гарантию

эффективности защиты). Эти три ключевых подпроцесса характерны для многих документов DO-XXX, включая DO-178, DO-254, DO-278 и т. д.; ниже они показаны применительно к кибербезопасности:

Группировка подпроцессов информационной безопасности

- 1. Планирование и область информационной безопасности** – стартуют первыми.
- 2. Процесс разработки мер защиты** – начинается после планирования.
- 3. Оценка рисков и гарантия эффективности** – выполняются непрерывно в проекте.

Для аспектов информационной безопасности полезнее сгруппировать подпроцессы так: (1) область информационной безопасности и оценка рисков, (2) разработка решений информационной безопасности, (3) гарантия эффективности защиты, тогда как планирование относится к сертификации верхнего уровня.

Эти 3 подпроцесса дополнительно детализированы в DO-356A/ED-203A (и частично в DO-326A/ED-202A):

1. Область информационной безопасности и оценка рисков:
 - a. Определение области информационной безопасности
 - b. Идентификация и оценка условий угроз
 - c. Характеризация сценария угрозы
 - d. Оценка уровня угрозы
2. Разработка решений информационной безопасности:
 - a. Разработка архитектуры и мер защиты воздушного судна
 - b. Разработка архитектуры и мер защиты системы, которая далее может быть развернута до уровня подсистемы, изделия и т. д.
 - c. Разработка руководящих указаний по информационной безопасности для интегратора системы
 - d. Разработка руководящих указаний по информационной безопасности для эксплуатанта воздушного судна
3. Гарантия эффективности защиты – 118 мероприятий в рамках 39 целей, включенных в 13 разделов, двух типов:
 - a. Специфическая гарантия информационной безопасности – включает 62 мероприятия в рамках 15 целей, включенных в 5 разделов
 - b. Гарантия разработки средств информационной безопасности – включает 56 мероприятий в рамках 24 целей, включенных в 8 разделов

**Специфическая
гарантия ИБ:**

**15 целей -> 62
мероприятия**

**Гарантия
разработки
средств ИБ:
24 цели -> 56
мероприятий**

1	• (5 целей -> 28 мер.)	Оценка рисков ИБ
2	• (2 цели -> 11 мер.)	Идентификация уязвимостей
3	• (3 цели -> 6 мер.)	Опровержение ИБ
4	• (1 цель -> 10 мер.)	Развертывание ИБ
5	• (4 цели -> 7 мер.)	Сохранение эффективности
6	• (5 целей -> 9 мер.)	Требования
7	• (3 цели -> 7 мер.)	Проектирование
8	• (3 цели -> 6 мер.)	Реализация
9	• (3 цели -> 10 мер.)	Верификация ИБ
10	• (2 цели -> 9 мер.)	Планирование ИБ
11	• (4 цели -> 8 мер.)	Управление конфигурацией
12	• (2 цели -> 5 мер.)	Связь по сертификации
13	• (2 цели -> 2 мер.)	ИБ инструментов

Гарантия информационной безопасности по DO-356A / ED203A: 39 целей, 118 мероприятий

Критически важный аспект информационной безопасности – время: это не просто однонаправленный процесс типа функциональной безопасности с четкой конечной точкой, после которой остается лишь мониторинг, а непрерывная борьба против любых потенциальных атак и атакующих, которые продолжают развиваться, даже если с самой системой больше ничего не происходит. Поэтому любая модификация, добавление, удаление или даже измененная функция должны повторно проходить обработку в соответствии с комплектом DO-326/ED-202, чтобы оценить, что изменилось со временем. Более того, даже само течение времени, без каких-либо модификаций системы или ее функций, может привести к изменениям в степени враждебности киберсреды, поэтому следование комплекту DO-326/ED-202 потребует периодического анализа рисков, чтобы оценить, остается ли система такой же защищенной, как предполагалось изначально. В этом аспекте необходимо следовать как “базовым” документам фазы разработки DO-326/ED-202 и DO-356/ED-203, так и документу эксплуатационной фазы DO-355/ED-204 по поддержанию летной годности.

Как можно эффективно выполнить руководящие материалы и рекомендации комплекта DO-326/ED-202?

Разумеется, даже самое подробное описание процессов информационной безопасности само по себе не гарантирует успеха. Более того, оно еще не решает вопрос эффективности; DO-326A/ED-202A прямо заявляет: “... понятие эффективности, определяемое как соотношение между полученными результатами и задействованными ресурсами, в данном стандарте не рассматривается”. Куда уж яснее...

Итак, что должен делать заявитель, чтобы удерживать сроки и стоимость этого процесса в разумных пределах?

Для гарантии эффективности защиты DO-356A/ED-203A определяет два метода ограничения требуемых усилий:

1. До примерно 25 % мероприятий по гарантии не являются “специфичными для информационной безопасности”, поэтому они могут быть удовлетворены предоставлением свидетельств того, что заявитель применял “обычные” практики гарантии разработки по функциональной безопасности, такие как ED79A/ARP4754A, DO-178C/ED-12C, DO-254/ED-80 и т. д.
2. Требования к мерам защиты зависят от уровня риска системы, поэтому требуются разные уровни гарантии. Уровни гарантии, присваиваемые соответствующим мерам защиты, обозначаются как уровень гарантии информационной безопасности (Security Assurance Level, SAL), по смыслу аналогичный уровням гарантии разработки функции и изделия (Function Development Assurance Level / Item Development Assurance Level, FDAL/IDAL) из параллельного процесса гарантии по функциональной безопасности. SAL 3 – наивысший уровень гарантии, SAL 0 означает отсутствие гарантии; нужно лишь доказать, что это действительно нулевой уровень. В результате надлежащее выполнение анализа рисков и назначение подходящих уровней гарантии означает, что усилия по обеспечению гарантии будут менее жесткими для систем и изделий, не отнесенных к SAL 3. Уровни гарантии приведены на следующем рисунке:

Уровень гарантии информационной безопасности (Security Assurance Level, SAL) для мер защиты

- **SAL 3** – наивысшая гарантия информационной безопасности.
- **SAL 2** – повышенная гарантия информационной безопасности; отличается от SAL 3 главным образом гарантией разработки мер безопасности.
- **SAL 1** – минимальная гарантия информационной безопасности; применяется для дополнительной защиты или усиления робастности.
- **SAL 0** – защитный эффект отсутствует; уровень ограничен потребностями защиты при исходной оценке.

Для оценки рисков и разработки решений информационной безопасности существуют различные методы, и даже DO-356A/ED-203A, например, предлагает четыре разных варианта оценки рисков, которые могут быть приемлемыми. Общий подход комплекта DO-326/ED-202 таков: “может существовать более одного приемлемого метода определения соответствия”, поэтому заявитель должен тщательно определить надлежащий метод с учетом специфики своего случая. Более того, комплект DO-326/ED-202 признает, что информационная безопасность летной годности – это “незавершенная работа”, поэтому к процессу могут подключаться появляющиеся меры, методы, стандарты и даже методологии для его упрощения.

Последний фактор, который может существенно повлиять на эффективность процесса информационной безопасности летной годности, – это уровень его интеграции с параллельным процессом обеспечения летной годности по функциональной безопасности. Одна сторона этого сходства проста: процессы действительно похожи, поэтому можно использовать входные данные процесса функциональной безопасности и извлекать из него уроки, сокращая усилия. Другой аспект, который необходимо учитывать, – степень объединения процессов. Подход комплекта DO-326/ED-202 здесь прост: “зависит от обстоятельств”: у тесной связки функциональной и информационной безопасности есть преимущества, например исключение повторения задач и снижение усилий на интеграцию, но есть и цена такой тесной связки, например: если функциональная безопасность требует после изменения конфигурации как можно меньшего числа изменений, то информационная безопасность может требовать очень частых изменений для учета меняющихся угроз, поэтому эти аспекты могут противоречить друг другу, если интегрированы в одно и то же изделие.

Итоги и вывод: параноики живут дольше

Пассажир заходит на борт коммерческого авиалайнера с ноутбуком, взламывает его сеть и заставляет его лететь еще выше: крайне маловероятно при использовании комплекта DO-326/ED-202. Основные выводы после этого краткого вводного обзора:

1. Авиационная кибербезопасность...

- a. ...является самостоятельной дисциплиной – она требует уникальной экспертизы, а также собственных методов анализа и соображений гарантии;
- b. ...затрагивает множество заинтересованных сторон, как внутри организации заявителя, так и вне ее, с которыми должно быть установлено взаимное доверие, а не просто предполагаться;
- c. ...это “бесконечная история”, поскольку киберугрозы продолжают развиваться со временем – поэтому информационная безопасность летной годности должна реализовываться как непрерывный процесс на протяжении всего срока службы всех сертифицированных воздушных судов и другого оборудования.

2. Комплект DO-326/ED-202...

- a. ...уже представляет собой пригодный для применения комплект документов, рассматриваемый как приемлемый метод определения соответствия (АМС) для авиационной кибербезопасности в США и Европе и быстро становящийся таковым в остальном мире;
- b. ...следует воспринимать как единое целое, а не “документ за документом”;
- c. ...опирается на ряд серьезных справочных источников в областях кибербезопасности и авиационной функциональной безопасности, чтобы создать целую “экосистему”, до такой степени, что некоторые методы и решения комплекта DO-326/ED-202 напрямую выведены из этих источников, тем самым предоставляя несколько конкретных вариантов, которые должны применяться в зависимости от конкретной организации, проекта и системы;
- d. ...все еще является развивающейся “незавершенной работой”, поэтому следует избегать “высекания в камне” каких-либо конкретных решений.

3. Процесс информационной безопасности летной годности...

- a. ...является ядром комплекта DO-326/ED-202, из которого черпают все документы комплекта;
- b. ...включает 7 шагов, 14 мероприятий, 62 цели – поэтому его нельзя рассматривать как нечто второстепенное в сертификации летной годности;
- c. ...в значительной степени опирается на интегральные процессы, а именно на “гарантию эффективности защиты”, включающую 118 мероприятий в рамках 39 целей, применяемых в зависимости от нескольких переменных, главным образом “уровня гарантии информационной безопасности”, который определяет требуемый уровень мер защиты для рассматриваемого элемента;
- d. ...по своей природе похож на процесс обеспечения летной годности по функциональной безопасности, поскольку оба используют оценки опасностей и рисков, тяжесть отказа и угрозы, требования к снижению риска и сходные методы гарантии, и это сходство предоставляет богатые возможности взаимной пользы для двух процессов, включая использование до примерно 25 % свидетельств гарантии по функциональной безопасности в качестве свидетельств гарантии разработки средств информационной безопасности;
- e. ...однако наиболее экономически эффективный уровень интеграции между процессами обеспечения летной годности по информационной и функциональной безопасности зависит от множества конкретных переменных, включая организационные факторы, тип оборудования и т. д., поэтому его лучше определять в каждом случае отдельно.

Поскольку эти пункты дают в сумме 12 основных выводов, необходимо добавить еще один вывод, чтобы получилось ровно 13, поэтому уместно завершить разумным советом г-на Педро Бустаманте, вице-президента по технологиям компании Malwarebytes, который, пожалуй, может быть самым важным выводом:

“Если вы не параноик, вы не выживете.”

Инженерные переходы

Автор: Вэнс Хилдерман

Рецензенты:

- Кевин Кендрина, руководитель направления инженерии качества, General Atomics
- Лиза Уинн (Дункан), инженер по качеству ПО, магистр искусств (Master of Arts, M.A.), сертифицированный инженер по качеству ПО (Certified Software Quality Engineer, CSQE).

Когда безопасность критична, критически важные с точки зрения безопасности системы должны обеспечивать упорядоченное и взвешенное выполнение инженерных работ, а также постоянное внимание к деталям. Практически у каждого инженерного мероприятия есть критерии входа и выхода. Их нужно заранее определить и документировать вместе с соответствующими критериями продукта и процесса, по которым сертифицирующий орган будет оценивать и подтверждать соблюдение требований. Например, прежде чем реализовывать логику ПО, то есть писать код, для критически важной системы уже должны быть подготовлены заранее определенные и формально рассмотренные артефакты под формальным управлением конфигурацией: требования по безопасности, системные требования, требования к ПО, проектные данные и стандарты кодирования ПО, по которым затем можно верифицировать код. Поэтому порядок инженерных мероприятий строится на переходах – небольших проверках на контрольных рубежах. Эти переходы должна оценивать гарантия качества (Quality Assurance, QA), а иногда и утверждать сертифицирующий орган, например Федеральное управление гражданской авиации США (Federal Aviation Administration, FAA), Агентство Европейского союза по авиационной безопасности (European Union Aviation Safety Agency, EASA) или военная приемка, прежде чем команда разработки сможет перейти к следующему этапу.

Инженерные переходы могут быть описаны в отдельном документе или включены в применимый План переходов ПО (Software Transition Plan, STP). В этом случае STP является дополнительным процессным документом и задает шаги переходов в инженерном жизненном цикле, выполняемые при планировании, разработке, верификации и валидации (Verification and Validation, V&V). Хотя такие переходы требуются авиационными инженерными руководствами, например DO-178 для бортового ПО, DO-254 для бортовой аппаратуры и DO-278 для наземных и космических систем связи, навигации, наблюдения и организации воздушного движения (Communication, Navigation, Surveillance / Air Traffic Management, CNS/ATM), многие компании не создают отдельный STP, а включают переходы в План разработки ПО (Software Development Plan,

SDP) и План верификации ПО (Software Verification Plan, SVP). Поскольку гарантия качества и гарантия процесса обычно отвечают за аудит и оценку соблюдения критериев перехода, План гарантии качества ПО должен описывать такие аудиты.

Планирование переходов нужно для того, чтобы показать: для инженерных переходов заранее определены критерии входа и выхода, эти критерии соблюдаются и их соблюдение проверяется аудитом. Например, требования к ПО должны быть рассмотрены и переведены в базовую версию до начала проектирования ПО высокого уровня. Следующие элементы, то есть критерии входа, должны быть доступны до проведения соответствующего рассмотрения требований к ПО:

Пример: критерии входа перед рассмотрением требований к ПО, переход

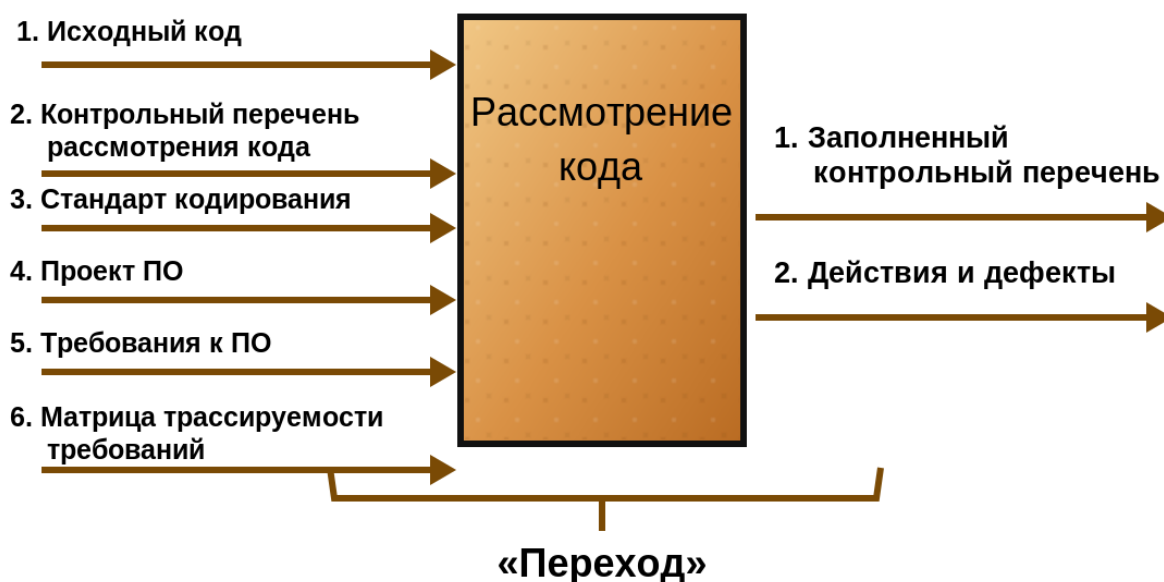
1. Рассмотренный и помещенный в базовую версию План разработки ПО и четыре других плана ПО для DO-178/DO-278.
2. Рассмотренный и помещенный в базовую версию стандарт требований к ПО.
3. Рассмотренные и помещенные в базовую версию требования к системе и безопасности, обычно соответствующие рекомендуемой аэрокосмической практике SAE ARP4754A.
4. Рассмотренные и помещенные в базовую версию требования по безопасности, соответствующие рекомендуемой аэрокосмической практике SAE ARP4761A.
5. Рассмотренная и помещенная в базовую версию матрица трассируемости, показывающая распределение требований к системе и безопасности на ПО.
6. Требования к ПО, помещенные в базовую версию для рассматриваемых требований.
7. Перечень контрольных вопросов рассмотрения требований к ПО, помещенный в базовую версию.

Соответствующее рассмотрение требований к ПО можно начинать только после того, как доступны все семь перечисленных выше элементов. После завершения рассмотрения требований к ПО должны быть завершены следующие элементы, то есть критерии выхода:

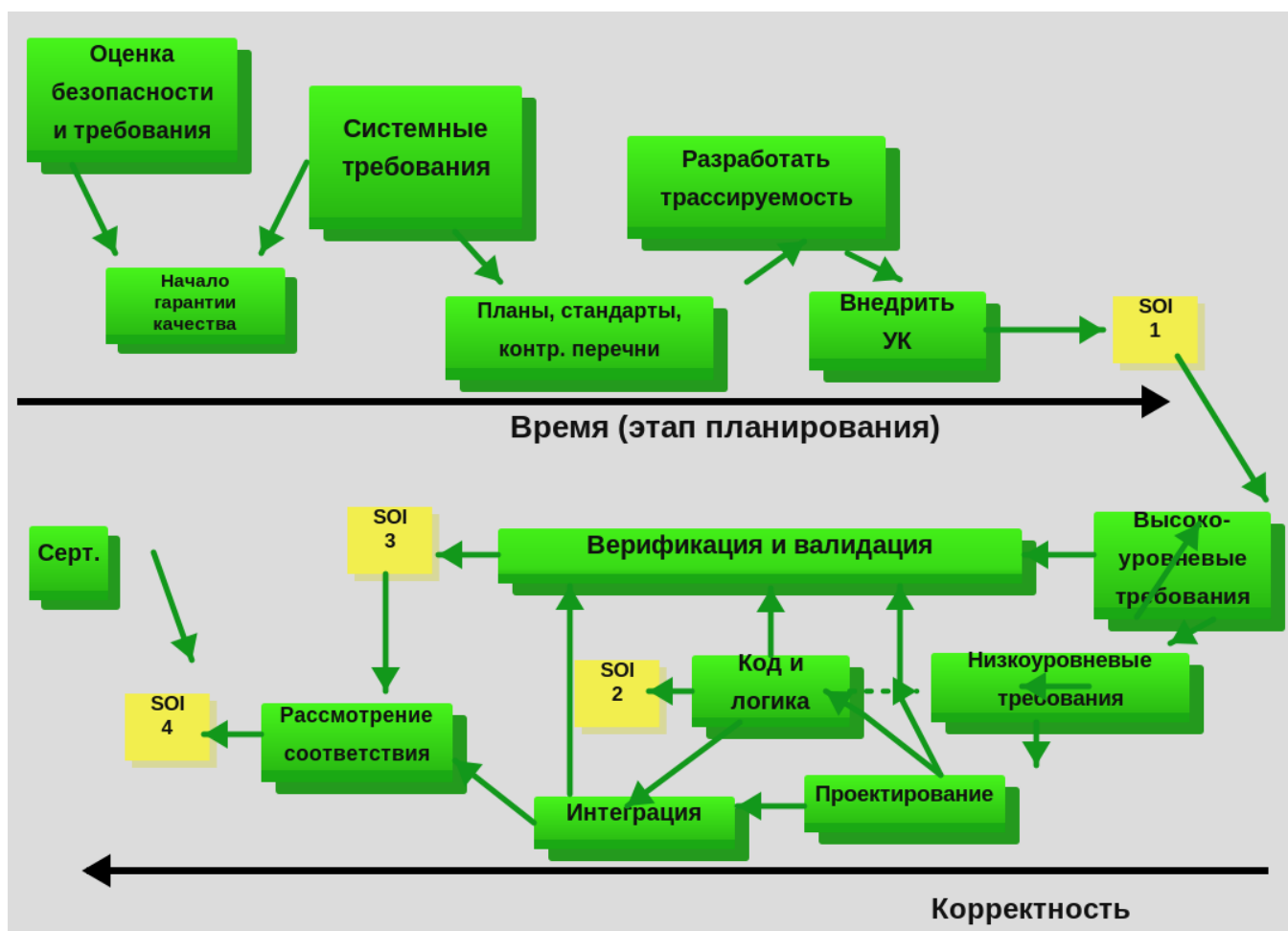


Пример: критерии выхода после рассмотрения требований к ПО, переход

В качестве еще одного примера рассмотрим официальное рассмотрение кода ПО или аппаратной логики, результаты которого засчитываются при сертификации. Для рассмотрения кода, выполняемого по DO-178 или DO-278, следующая диаграмма показывает входные и выходные артефакты такого рассмотрения. Каждый из них должен находиться под управлением конфигурацией:



Таким образом, приведенные выше входные и выходные артефакты образуют один переход. Этот процесс повторяется для всех инженерных переходов по DO-178 и DO-278, но для каждого перехода определяются свои критерии входа и выхода. Теперь рассмотрим разные переходы на оптимальном инженерном маршруте по DO-178 и DO-278. Каждая стрелка ниже представляет один переход:



Оптимальный инженерный маршрут с переходами, по Вэнсу Хилдерману

На предыдущем рисунке каждое ключевое инженерное мероприятие представлено блоком, а стрелки-переходы соединяют эти блоки от начала до конца. С каждой такой стрелкой связаны уникальные, заранее определенные входные задачи, мероприятия по верификации и критерии выхода. Эти критерии должны быть выполнены, чтобы мероприятие считалось проведенным надлежащим образом. Но можно возразить: если процесс соблюден и критерии перехода выполнены, разве это доказывает совершенство продукта? Нет. Процессные критерии перехода предназначены не для доказательства совершенства продукта. Оценка переходов относится именно к процессу: сначала заранее обдумываются и формально определяются критерии входа и выхода, затем оценивается, насколько инженерная команда им следовала. Инженер по верификации должен использовать входные и выходные артефакты перехода при выполнении своей работы; затем персонал гарантии качества или гарантии процесса оценивает работу этого инженера и подтверждает, что критерии перехода соблюдались и есть соответствующие свидетельства. Важно помнить: эфемерного совершенства продукта не существует даже в авиации. Вместо этого экосистема оценок безопасности удерживает неизбежные несовершенства в приемлемых пределах риска. Кроме того, инженерные рассмотрения предназначены для оценки технических характеристик продукта, а гарантия качества

выполняет оценки переходов, чтобы предоставить сертифицирующему органу независимые объективные свидетельства процессного соответствия авиационным руководствам, таким как DO-178, DO-254, DO-278 и другие.

Этапы взаимодействия с сертифицирующим органом (SOI)

Ниже показаны четыре этапа взаимодействия с сертифицирующим органом (Stage of Involvement, SOI), обычно применяемые в авиационной инженерной деятельности. Эти этапы образуют четыре основных контрольных рубежа для оценки достижения переходов верхнего уровня. Этапы SOI основаны на процессе и, как правило, проводятся под руководством гарантии качества или гарантии процесса.

Краткое описание: основное назначение каждого этапа взаимодействия с сертифицирующим органом (Stage of Involvement, SOI)

1. **Этап SOI-1:** планирование завершено? Готовы перейти к реализации?
2. **Этап SOI-2:** реализация завершена? Готовы перейти к верификации?
3. **Этап SOI-3:** верификация завершена? Готовы перейти к рассмотрению соответствия?
4. **Этап SOI-4:** соответствие подтверждено? Готовы подтвердить сертифицируемость?

Более подробное представление SOI приведено на следующем рисунке. Он обобщает ключевые мероприятия, связанные с каждым этапом:

Представление каждого этапа взаимодействия с сертифицирующим органом (Stage of Involvement, SOI) на среднем уровне

Этап SOI-1: области внимания

- Рассмотрение оценки безопасности.
- Системные требования.
- Готовые коммерческие и повторно используемые компоненты.
- Предлагаемые альтернативные методы определения соответствия (Alternative Means of Compliance, AMC).

Этап SOI-2: области внимания

- Трассировка требований высокого уровня (High-Level Requirements, HLR).
- Детализация требований низкого уровня (Low-Level Requirements, LLR) для уровня гарантии разработки С и выше (Development Assurance Level, DAL).
- Инженерные контрольные перечни.
- Управление конфигурацией и гарантия качества при повторном рассмотрении.

Этап SOI-3: области внимания

- Структурное покрытие к требованиям для уровня гарантии разработки С и выше.
- Архитектура и связность.
- Тестирование на робастность.
- Метрики дефектов.

Этап SOI-4: области внимания

- Покрытие тестами.
- Выборочная цепочка сообщений о проблемах (Problem Report, PR).
- Необоснованные изменения.
- Свидетельства изменений.
- Открытые сообщения о проблемах.

Подробности SOI-1

SOI-1 похож на проверку строительных планов перед возведением офисного здания рядом с сейсмическим разломом: правильно ли оценены факторы безопасности и будет ли здание безопасным, если строительство выполнено по этим планам? В авиации SOI-1 включает следующие ключевые вопросы. Ответы на них должны быть ясными и поддающимися верификации:

- Начаты ли оценка функциональных опасностей (Functional Hazard Assessment, FHA), предварительная оценка безопасности системы (Preliminary System Safety Assessment, PSSA) или предварительная оценка безопасности воздушного судна (Preliminary Aircraft Safety Assessment, PASA) и анализ общих причин так, чтобы риски были выявлены, а предлагаемый уровень гарантии разработки (Development Assurance Level, DAL) был обоснован?
- Определены ли методы обеспечения соответствия рекомендуемой аэрокосмической практике ARP4754A (Aerospace Recommended Practice 4754A) для воздушного судна или системы и ARP4761A для безопасности, а также исходные требования по безопасности и требования к воздушному судну или системе?
- Завершены ли все необходимые планы и стандарты в соответствии с применимыми и заданными нормативными требованиями?
- Определена ли архитектура воздушного судна или системы, включая использование готовых коммерческих изделий (Commercial Off-The-Shelf, COTS) и унаследованных элементов?
- Предлагаются ли какие-либо альтернативные методы определения соответствия (Alternative Means of Compliance, AMC), и есть ли обоснование их приемлемости?
- Определено ли, какие инженерные инструменты планируется использовать, и рассмотрена ли потребность в их квалификации?
- Если SOI-1 будет одобрен, будет ли достаточно безопасно начинать разработку воздушного судна или системы по этим планам?

Таким образом, SOI-1 обеспечивает переход к разработке, которая затем завершается SOI-2.

Подробности SOI-2

SOI-2 похож на инспекцию здания, когда его основные части уже построены: соответствовало ли строительство ранее утвержденным, например на SOI-1, строительным планам? SOI-2 включает следующие ключевые вопросы. Ответы на них должны быть ясными и поддающимися верификации:

- Следовала ли разработка требований, проекта и кода или логики ранее утвержденным на SOI-1 планам, стандартам и критериям перехода, что подтверждено проверенными аудитом свидетельствами верификации?
- Имеются ли требования, включая системные требования и требования по безопасности, требования к ПО высокого и низкого уровней, а также требования к аппаратуре, с двунаправленной трассируемостью и записями верификации, подтверждающими процессную декомпозицию и уточнение?
- Имеются ли записи рассмотрений и аудитов, показывающие, что требования, проект и код или логика были разработаны в соответствии с заранее определенными переходами, описанными выше, включая контрольные перечни для каждого рассмотрения?
- Имеются ли записи управления конфигурацией, показывающие, что управление артефактами было введено, все входы и выходы переходов зафиксированы, а все изменения после первоначальных рассмотрений сопровождалось зарегистрированными повторными рассмотрениями?
- Выполнялись ли мероприятия гарантии качества или гарантии процесса независимо, и выполнялись ли соответствующие мероприятия по верификации для артефактов DAL A и DAL B независимо, как это требуется?

Таким образом, SOI-2 обеспечивает переход к верификации, которая затем завершается SOI-3.

Подробности SOI-3

SOI-3 устанавливает, что мероприятия по валидации и верификации были выполнены в соответствии с планами и стандартами. SOI-3 включает следующие ключевые вопросы. Ответы на них должны быть ясными и поддающимися верификации:

- Были ли рассмотрения и тестирование применены ко всему ПО в соответствии с уровнем гарантии разработки (DAL), включая трассируемость, критерии перехода и анализ структурного покрытия, то есть кода и требований, для DAL C, DAL B и DAL A?
- Являются ли архитектура и проект детерминированными, что подтверждено тестированием на основе требований, а также оценкой потоков данных, оценкой потоков управления и анализом межкомпонентной связности?

- Было ли тестирование на робастность выполнено и рассмотрено в достаточном объеме?
- Были ли дефекты, обнаруженные в процессе верификации, классифицированы и закрыты?
- Дополнительно: анализировались ли метрики дефектов для определения достаточности связанных процессов?

Таким образом, SOI-3 обеспечивает переход к рассмотрению соответствия, которое затем завершается SOI-4.

Подробности SOI-4

SOI-4 устанавливает, что предыдущие процессы (SOI-1, SOI-2 и SOI-3) выполнялись на протяжении всего инженерного жизненного цикла в соответствии с заявленными критериями, а мероприятия по валидации и верификации были завершены согласно планам и стандартам. SOI-4 включает следующие ключевые вопросы. Ответы на них должны быть ясными и поддающимися верификации:

- Были ли все изменения в планах, стандартах и контрольных перечнях утверждены и документированы, включая обоснование для элементов, которые не соответствуют последней редакции этих документов?
- Выполнялись ли все изменения по заданным процессам управления конфигурацией?
- Были ли все нерешенные проблемы подвергнуты анализу безопасности, в котором подробно описаны потенциальное влияние на безопасность и эксплуатационные ограничения?
- Есть ли достаточные свидетельства необходимых аудитов гарантии качества или гарантии процесса с подтверждением устранения несоответствий?
- Достаточно ли полон указатель конфигурации, чтобы в будущем можно было оценивать артефакты, вносить изменения и выполнять повторную верификацию эксплуатационного изделия, включая воссоздание идентичного изделия в течение его срока эксплуатации?
- Есть ли в итоговом отчете о выполнении работ (Accomplishment Summary) заявление о соответствии применимым стандартам и руководствам?

Таким образом, переходы являются одним из краеугольных камней базовой сертификации авиационной разработки. Важно планировать и документировать подробные переходы для инженерных процессов, но не менее важно независимо оценивать такие переходы силами гарантии качества и вести соответствующие записи. Автор слишком часто видел, как в целом хорошие авиационные проекты не проходили оценку переходов у сертифицирующего органа и были вынуждены тратить несоразмерные ресурсы, возвращаясь к исходной точке.

Трассируемость в авиационной разработке

Автор: Вэнс Хилдерман

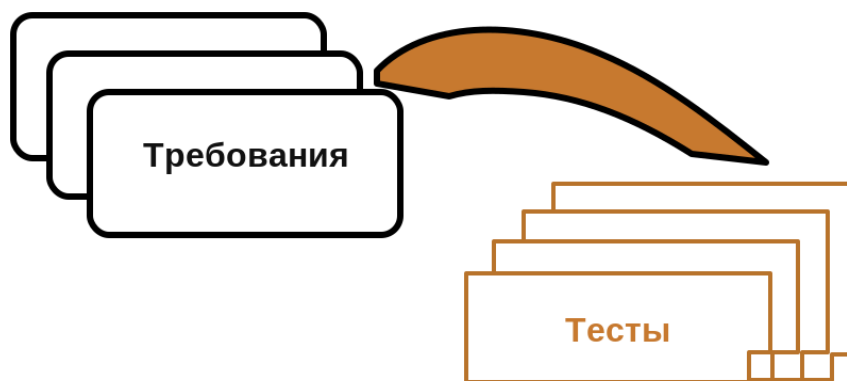
Рецензенты:

- г-н Bill De Leeuw, уполномоченный представитель Федерального управления гражданской авиации США (Federal Aviation Administration, FAA) по системам, оборудованию и ПО
- г-н Kevin Kendryna, руководитель направления инженерии качества, General Atomics
- г-жа Lisa Wynn (Duncan), инженер по качеству ПО, магистр искусств (Master of Arts, M.A.), сертифицированный инженер по качеству ПО (Certified Software Quality Engineer, CSQE).

Трассируемость в авиационной разработке: краеугольный камень

Трассируемость (Traceability) является важным элементом большинства современных профессий. Ее используют бухгалтеры, исследователи и судебные эксперты. Однако для качественной инженерной разработки критически важных с точки зрения безопасности систем трассируемость становится не просто условием успеха, а способом доказать этот успех при сертификации.

Большинство современных стандартов системной инженерии, разработки ПО, а теперь и аппаратуры, требуют трассируемости в той или иной степени. Почему? Трассируемость – понятный способ убедиться, что реализация соответствует спецификации. Традиционно она означала всего лишь проверку того, что каждому уникальному требованию соответствуют тестовые примеры, подтверждающие его реализацию. При таком упрощенном подходе достаточно разработать тесты для каждого требования, а затем связать, или “протрассировать”, эти тесты с данным требованием. Когда все требования таким образом были связаны с конкретными тестовыми примерами, трассируемость и, скорее всего, само тестирование считались завершенными, как показано на следующем рисунке:



Упрощенная трассируемость: начало

Упомянутый выше упрощенный подход к трассируемости действительно полезен и является первым шагом в правильном направлении. Но полон ли он? Едва ли. Соответствует ли он DO-178C или DO-254, которые, пожалуй, относятся к самым строгим в мире стандартам для авионических систем? Конечно нет. В чем недостатки такого упрощенного подхода?

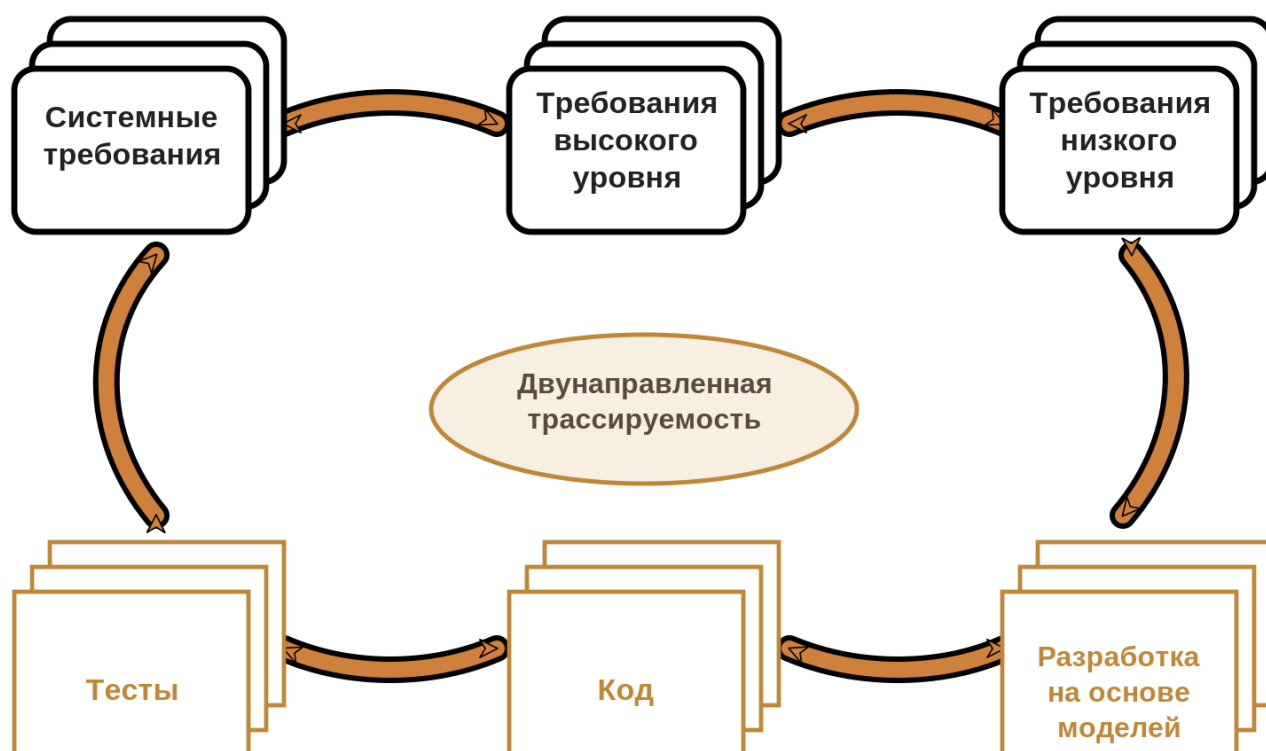
- Нет трассируемости к компонентам реализации, а она необходима при рассмотрении этих компонентов.
- Нет трассируемости между уровнями детализации требований, а она необходима для доказательства процесса декомпозиции требований.
- Нет трассируемости к проектным данным, а она необходима при их рассмотрении.
- Нет возможности выявлять лишние компоненты реализации, например неиспользуемый код.
- Трассируемость является однонаправленной, а не двунаправленной.

Чтобы устранить эти недостатки, DO-XXX и ARP47XX требуют более детальной и полной трассируемости, которая ведется не только вперед, но и назад, то есть является двунаправленной. Кроме того, DO-178C, DO-278A и DO-254 требуют полной трассируемости по всему набору артефактов, включая:

- системные требования;
- требования к ПО или аппаратуре высокого и низкого уровней;
- реализацию: код или язык описания аппаратуры VHDL (VHSIC Hardware Description Language, VHDL);
- тесты.

Все перечисленное фиксируется в указателе конфигурации (Configuration Index, CI).

Эта более полная трассируемость, применимая к ПО по DO-178C и DO-278A, показана на рисунке ниже. Обратите внимание: блок “разработка на основе моделей” (Model-Based Development, MBD) можно заменить обычным проектом без применения разработки на основе моделей, и тогда это будет просто “проект”, если так уместнее:



Современная авиационная трассируемость: результат эволюции (показана разработка на основе моделей)

Как видно из рисунка, трассируемость по DO-XXX и ARP47XX по сути образует замкнутый контур и является двунаправленной. Контур замкнут потому, что трассируемость ведется сверху вниз, от требований к коду и тестам, и снизу вверх, от кода и тестов к связанным требованиям. Кроме того, трассируемость должна поддерживать связи “один ко многим” и “многие к одному”. Что это означает? Все просто: один элемент трассируемости может быть связан с несколькими элементами нижнего уровня, а несколько элементов верхнего уровня могут быть связаны с одним и тем же элементом нижнего уровня. Например, одно системное требование может быть, и обычно бывает, декомпозировано на несколько требований нижнего уровня; поэтому это системное требование трассируется к каждому из них. И наоборот, для верификации одного требования к ПО может потребоваться несколько тестовых примеров; каждый из них трассируется обратно к этому единственному требованию и вместе с остальными обеспечивает его совокупное покрытие.

В критически важных с точки зрения безопасности системах безопасность полетов является очевидным приоритетом. Обычно требования, связанные с безопасностью полетов, выявляются в процессах анализа безопасности, регламентированных ARP-4761A и ARP-4754A. Эти требования имеют первостепенное значение и должны учитываться на протяжении инженерного жизненного цикла, а затем и в эксплуатации. Поэтому требуется трассируемость от них к требованиям нижнего уровня, которые помогают реализовать меры безопасности или связаны с ними.

Каждый раз, когда по требованию, связанному с таким требованием безопасности, выполняется какая-либо работа, например изменение или верификация, должна срабатывать обратная связь с процессом оценки безопасности. Трассируемость является ключевым механизмом, который запускает и фиксирует такие связи с безопасностью.

Традиционная отраслевая трассируемость, в отличие от трассируемости по DO-XXX, является односторонней: она ведется сверху вниз и подтверждает, что каждое требование реализовано и верифицировано. Во многих отраслях такой нисходящей трассируемости достаточно. Однако в критически важных с точки зрения безопасности отраслях, особенно при применении DO-XXX, ее недостаточно. Почему? Потому что высококачественная система должна гарантировать отсутствие лишней реализации и лишнего тестирования. Например, неиспользуемый и недокументированный код – беда для сопровождения изделия и гарантии качества: он может приводить к неверному пониманию и ошибочным предположениям о поведении кода, особенно при рассмотрении кода или модернизации изделия. В итоге это повышает вероятность ошибок во время выполнения. Поэтому DO-XXX для аппаратуры и ПО требуют, чтобы у каждой функции кода было по меньшей мере одно трассируемое к ней требование. Это доказывает, что каждая функция присутствует в конечном изделии не случайно. Кроме того, контролируемая матрица трассируемости должна использоваться как входные данные для рассмотрений. Например, при рассмотрении кода ПО необходимо учитывать требования к ПО, распределенные на рассматриваемый код, согласно контролируемой матрице трассируемости.

Почему трассируемость?

Если спросить десять инженеров, какой самый важный вклад дает трассируемость, вероятно, будут получены десять разных ответов. На самом деле трассируемость дает много разных преимуществ. Системный инженер скажет, что она помогает убедиться в реализации всех заданных требований. Проектировщик скажет, что она обеспечивает корректное архитектурное распределение требований через оптимальный проект. Разработчик кода может отметить, что она помогает гарантировать отражение всех функциональных требований в коде. Специалист по тестированию скажет, что она помогает сопоставить каждое требование с тестовым примером или несколькими тестовыми примерами, которые его верифицируют. Специалист по гарантии качества скажет, что трассируемость позволяет убедиться, что все остальные инженерные

дисциплины сделали свою работу. Наконец, руководство инженерного подразделения скажет, что метрики трассируемости позволяют точно оценивать состояние проекта и не получать на вопрос о ходе работ ответ в стиле “я на 95 % закончил, шеф, честное слово”.

Поэтому в современных процессах разработки ПО и аппаратуры трассируемость используется, как минимум, для ответа на следующие вопросы:

1. *Где реализовано каждое требование?*
2. *Распределено ли каждое требование?*
3. *Соответствует ли реализация требованиям?*
4. *Связано ли требование с безопасностью полетов, и если да, обозначены ли связи трассируемости к связанным и родительским требованиям безопасности для непрерывного анализа влияния на безопасность?*
5. *Какие верификационные тестовые примеры будут использоваться для верификации данного требования?*
6. *Каково совокупное влияние изменения любого требования или кода?*
7. *Завершен ли проект и готов ли он к сертификации?*
8. *Действительно ли поставляемое изделие является тем, что изначально запросили и задали?*

Последний пункт интересен: дополнительная важная неявная цель трассируемости состоит в том, чтобы через указатель конфигурации и все действия по управлению конфигурацией доказать, что сконфигурированное изделие, поставляемое заказчику, действительно является тем изделием, которое было запрошено. Существует множество типов систем и инструментов трассируемости, но все они должны как минимум без затруднений отвечать на приведенные выше вопросы, чтобы считаться достаточными для выполнения сертификационных требований DO-XXX и ARP47XX.

Инструменты трассируемости

Теперь понятно, что полная трассируемость для критически важных с точки зрения безопасности систем не просто желательна, а обязательна. Но обязательны ли специализированные инструменты трассируемости? Нет. Современные инструменты не обязательны, как не обязательны современные инструменты для строительства дома. В мире до сих пор есть люди, у которых почти нет выбора, кроме как жить в домах, построенных из палок и грязи. Но в реальном мире, то есть в вашем мире инженерной разработки, инструменты трассируемости нужны для повышения точности, повторяемости и соблюдения графика. В конечном счете необходимо доказать, что трассируемость является и точной, и полной.

Результатом трассируемости должна быть возможность предоставлять данные трассировки между ключевыми компонентами проекта, включая требования, проект, реализацию и тесты. Теоретически, а также для очень малых проектов, трассируемость можно обеспечить электронной таблицей или простой таблицей внутри документа. Создание и сопровождение таких таблиц вручную трудоемко и подвержено ошибкам. В частности, ручная трассируемость плохо поддерживает обновления и быстрые аудиты гарантии качества. Однако для очень малых проектов она может быть приемлемой. Для несколько более крупных проектов, где больше 2 000-3 000 строк кода или больше 100 требований, инструмент трассируемости всегда оказывается экономически оправданным и дает экономию не менее чем в 5 раз по сравнению с ручной трассируемостью. Какие варианты таких инструментов существуют?

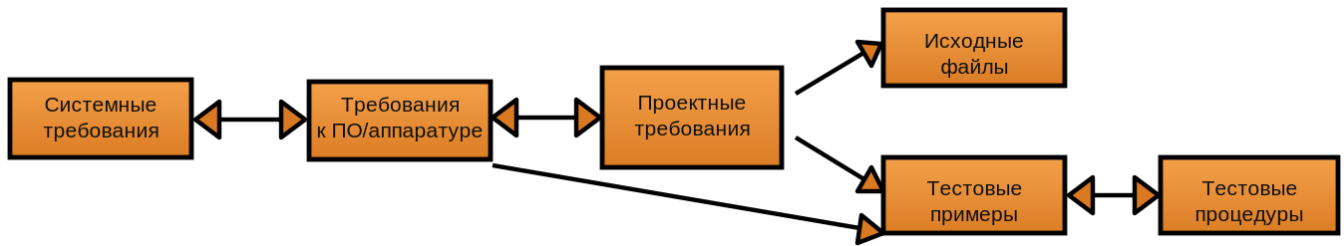
- а. создать собственный внутренний инструмент; или
- б. купить специализированный инструмент трассируемости, разработанный для DO-178C/254; или
- с. купить готовый коммерческий инструмент трассируемости (Commercial Off-The-Shelf, COTS), например DOORS, RequisitePro, Jama, JIRA и т. д.

Каковы базовые признаки приемлемого инструмента трассируемости? Независимо от выбранного инструмента следует убедиться, что он обеспечивает:

1. Гибкие механизмы тегирования, чтобы не приходилось менять ваши стандарты.
2. Двухнаправленную трассируемость, а также связи “один ко многим” и “многие к одному” между требованиями, проектом, кодом и элементами тестирования.
3. Автоматические обновления, чтобы избежать подверженной ошибкам ручной трассировки.
4. Автоматическое указание на отсутствующие элементы трассируемости.
5. Специфичную для данного инструмента возможность сопряжения с инструментом управления требованиями для последующей интеграции.
6. Совместимость с требованиями DO-XXX и ARP47XX, если вы разрабатываете авионику.

Нужно ли формально квалифицировать инструменты трассируемости по DO-330, как рассматривалось в одной из предыдущих глав этой книги? На первый взгляд кажется, что нужно, потому что они автоматизируют аспекты трассируемости, требуемые DO-XXX. Однако если выходные данные инструмента трассируемости проверяются вручную, квалификация не требуется. Поскольку большинство пользователей инструментов трассируемости действительно выполняют ручную проверку трассируемости в рамках этапа взаимодействия с сертифицирующим органом (Stage of Involvement, SOI), а именно SOI-4, квалификация инструмента трассируемости обычно не требуется и не выполняется.

В качестве последней проверки убедитесь, что ваш инструмент трассируемости поддерживает типовую базовую парадигму трассируемости, например показанную на рисунке ниже для ПО. Обратите внимание: аппаратура также требует трассируемости, и в идеале она должна поддерживаться тем же инструментом или общей базой данных.



Базовая парадигма структуры трассируемости систем, ПО и аппаратуры

Валидация и верификация авиационных систем (V&V)

Автор: Вэнс Хилдерман

Рецензент:

Кеннет Р. Хеберт, доктор философии (PhD), старший профильный эксперт по системам и процессам, AFuzion Inc.

Валидация и верификация, верификация и валидация или просто «валидация и верификация» (Validation & Verification, V&V)?

Валидация и верификация лежат в основе авиационной разработки и разработки ПО, критически важного с точки зрения безопасности. Фактически они являются ключевыми элементами любой успешной инженерной среды разработки. Если некоторый объект не был валидирован или верифицирован, откуда знать, что он собой в действительности представляет и можно ли на него полагаться? Во многих публикациях и отраслях эту аббревиатуру называют “Verification & Validation”, ставя “Verification” на первое место. Однако любой хороший инженер знает, что валидация как изначально предшествует верификации, так и важнее ее. Валидация дает представление о возможностях изделия, тогда как верификация оценивает его содержимое. Почему же тогда DO-178C (Software Considerations in Airborne Systems and Equipment Certification) и DO-278A (Software Integrity Assurance Considerations for Communication, Navigation, Surveillance and Air Traffic Management Systems) требуют верификации, но не валидации? Об этом чуть ниже.

Валидация и верификация образуют основу авиационного процесса обеспечения “корректности”. В самом простом определении применимо следующее:

- **Валидация:** *оценка степени того, насколько требования, связанные с объектом, корректны, однозначны, полны и верифицируемы.*
- **Верификация:** *оценка степени того, насколько реализация объекта соответствует предъявленным к нему требованиям.*

Валидация выполняется на протяжении инженерного жизненного цикла, но на начальном этапе она предшествует проектированию: требования нужно оценить и улучшить до проектирования и реализации. Это позволяет затем рассматривать проектирование и реализацию на соответствие связанным с ними требованиям. Верификация, напротив, выполняется после создания артефакта или объекта, чтобы оценить его корректность.

Надлежащая валидация до реализации может предотвращать дефекты. Верификация никогда не может предотвратить эти дефекты, а лишь обнаруживает их, чтобы затем можно было внести исправления. Исследователи в целом согласны с тем, что

предотвращение ошибок (включая валидацию) на 500 % – 1000 % экономически эффективнее, чем обнаружение этих ошибок посредством верификации. Следовательно, валидация сначала предшествует верификации, а затем оказывается еще и экономически выгоднее. Почему же тогда авиационные руководящие документы по ПО DO-178C и DO-278A, по-видимому, игнорируют валидацию и вместо этого сосредоточены на верификации? Потому что валидация включает оценку полноты требований, которую невозможно выполнить полностью, пока не интегрированы также аппаратура и система; до этого момента валидация на уровне ПО носит субъективный характер. Но рассмотрение требований к ПО жизненно важно и составляет основу DO-178C и DO-278A; это рассмотрение требований выполняется как мероприятие процесса верификации до проектирования ПО и тем самым воспроизводит часть валидации.

Валидация начинается с оценки требований к объекту и продолжается вплоть до реализации. В авиации валидация выполняется для воздушного судна, аппаратуры и системы, но не для ПО внутри системы. Почему? Из-за приведенного выше определения валидации. Валидация включает определение “полноты” требований. Но одно только ПО не может быть валидировано, поскольку такая валидация требует полностью интегрированных аппаратуры и ПО. Следовательно, “валидация” ПО по своей сути выполняется как часть валидации и верификации авиационной системы. Валидация должна охватывать как предусмотренную, так и непредусмотренную функциональность. Она выполняется с применением комбинации следующих методов, чтобы установить, является ли объект корректным, однозначным, полным и верифицируемым:

- Двухнаправленная трассируемость
- Анализ
- Моделирование (см. следующий раздел)
- Испытание
- Подобие

Инженерное рассмотрение

Верификация, в свою очередь, сосредоточена на оценке артефакта на соответствие его требованиям. “Уравнение верификации” для авиационного ПО показано ниже:

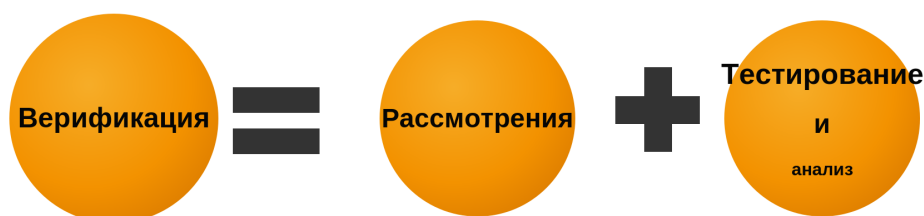


Иллюстрация уравнения верификации

Ниже кратко описаны рассмотрения, тестирование и анализ.

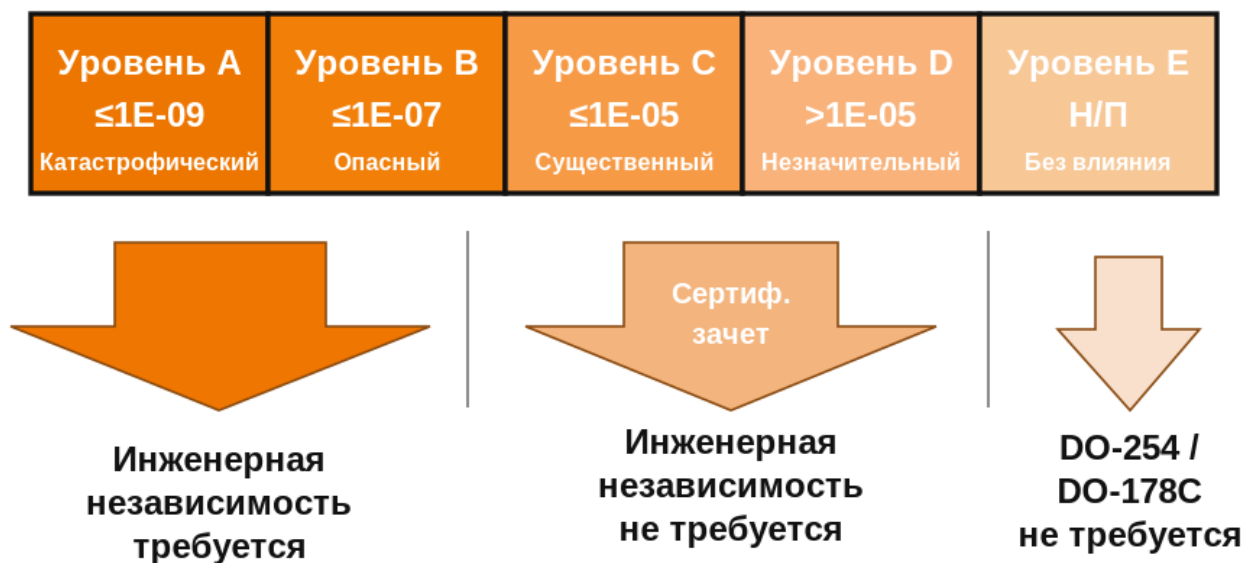
Рассмотрения

Как показано выше на иллюстрации уравнивания верификации, “Рассмотрения” (самый крупный шрифт) представляют наибольший объем работы, поскольку все артефакты, разрабатываемые авиационными инженерами в поддержку сертификации, подлежат рассмотрению. В частности, в случае авиационного ПО рассматриваются следующие ключевые авиационные артефакты:

- Планы и стандарты
- Требования к воздушному судну, системе, ПО и аппаратуре, а также соответствующие:
 - Проектные данные
 - Код
 - Тесты и результаты
 - Трассируемость
 - Сообщения о проблемах и исправления

Для более критичных артефактов рассмотрения должны выполняться независимо, если это требуется уровнем гарантии разработки (Development Assurance Level, DAL). Как правило, для выходных данных процессов уровня гарантии разработки А и В (например, “артефактов”, таких как требования, проектные данные, код, тесты и т. п.) требуется подтверждение независимого рассмотрения. Следующая диаграмма показывает применение независимости валидации и верификации в зависимости от различной критичности систем для крупных воздушных судов/систем:

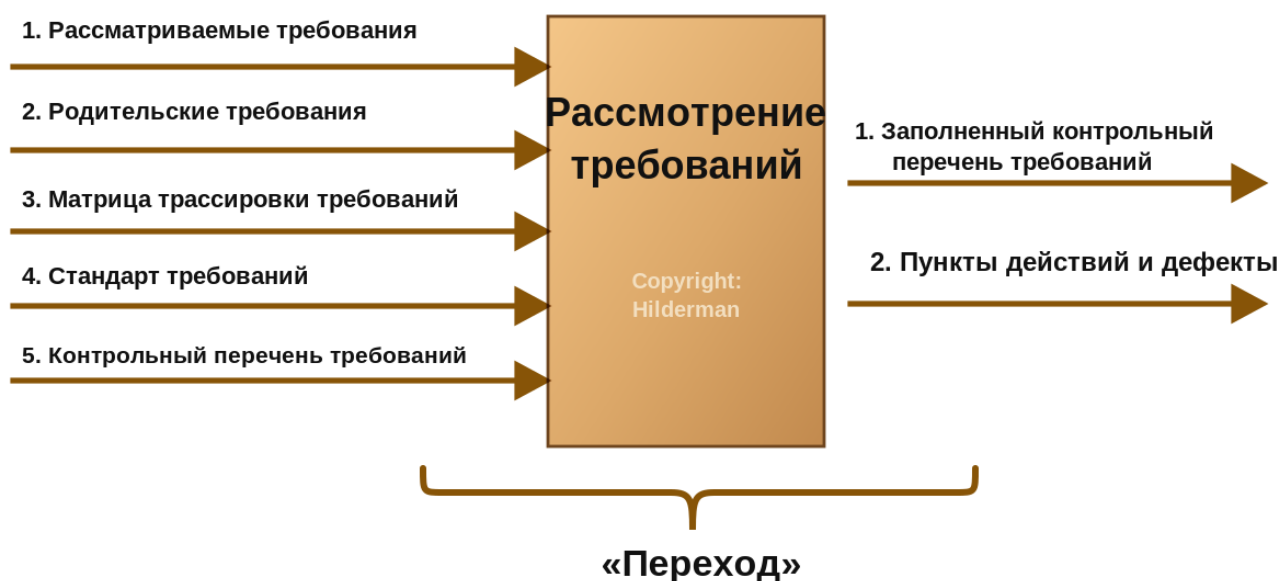
Инженерная «независимость» по уровню гарантии разработки (DAL). (Примечание: гарантия качества всегда независима):



На рисунке выше “независимость” означает другое лицо, применяющее несвязанный процесс. Очевидно, что отдельные лица независимы. Однако отдельные лица могут быть связаны и более тонким образом. Например, если Джон должен быть независимым

рецензентом разработки GPS-системы Мэри, Джону не разрешается полагаться на связанную (и потому, возможно, дефектную) GPS-модель Мэри. Одним из ключевых исключений в части независимости является верификация проектирования аппаратуры: поскольку проектирование аппаратуры по своей природе итеративно и сложно по многим причинам, включая температурные, ресурсные и компоновочные соображения, использование независимого лица для разработки тестов в действительности может дать менее тщательную верификацию. Поэтому для аппаратуры, в отличие от ПО уровня гарантии разработки А и В, разработчику разрешается тестировать собственный проект при условии, что затем эти тесты рассматривает независимое лицо.

Авиационные рассмотрения должны выполняться последовательно и с наличием свидетельств. Поэтому для рассматриваний предписывается заранее определенный набор входных и выходных данных, специфичный для типа рассматриваемого объекта. Например, для требований к аппаратуре и ПО требуется рассмотрение требований для всех уровней гарантии разработки, связанных с безопасностью полетов (то есть уровней А, В, С и D, но не Е). В случае требований к ПО применяются следующие заранее определенные входные и выходные данные, образующие “критерии перехода” (подробности о переходах см. в главе 17 настоящей книги).



На рисунке выше рецензенты требований используют все пять входных данных для выполнения рассмотрения. Все отмеченные поручения или дефекты затем возвращаются в процесс разработки требований, после чего рассмотрение повторяется для проверки корректности обновления. Рассмотрения требований часто приводят к обновлению родительских требований для их уточнения, а также оценивают корректность обязательной трассируемости к родительским требованиям, поэтому синхронизация между входами и выходами имеет первостепенное значение и полностью отслеживается в рамках процессов верификации и управления конфигурацией. Затем служба гарантии качества (Quality Assurance, QA) проводит аудит этих рассмотрений, чтобы удостовериться, что ведение записей подтверждает правильное применение входных и выходных данных в ходе процесса рассмотрения; это называется аудитом критериев

перехода. Все прочие упомянутые ранее рассмотрения артефактов аналогичным образом требуют критериев перехода и соответствующего ведения записей. Образец контрольного перечня рассмотрения требований к ПО см. в приложении В к этой книге.

Как обычно в авиации, валидация и верификация требуют определенного планирования до начала фактических мероприятий, чтобы последующие рассмотрения этих мероприятий можно было оценить на согласованность с этим предварительным планированием. Чтобы валидация и верификация обеспечивали необходимую степень доверия, необходимо учитывать огромное количество деталей и принимать бесчисленные решения, уравнивающие стоимость и получаемую информацию. Они могут быть настолько сложными и тонкими, что требуют систематического планирования (как и любое сложное и ресурсоемкое мероприятие). Как минимум, при планировании валидации следует рассмотреть следующую информацию (через объективные критерии оценки):

Темы планирования валидации

- Процессы обеспечения полноты заданных требований и соответствия потребностям пользователя.
- Определение структурированного процесса валидации, включая организационные интерфейсы и независимость валидации.
- Задание применяемых методов валидации, собираемых данных, порядка их регистрации и доступа к ним.
- Порядок обновления валидации при изменении требований.
- График мероприятий по валидации.

Верификация на уровне системы (обычно выполняемая в авиации по рекомендуемой аэрокосмической практике ARP4754A (Aerospace Recommended Practice, ARP)) обобщена на следующем рисунке и далее рассматривается ниже:

Методы верификации на уровне системы

1. **Инспекция или рассмотрение:** визуальная проверка артефактов на соответствие требованиям.
2. **Оценка с применением моделей:** изучение моделей или их элементов для оценки поведения системы в реальном времени.
3. **Анализ:** подробное исследование инженерных данных по критериям приемки.
4. **Испытание или демонстрация:** работа системы для сопоставления фактических результатов с ожидаемыми.
5. **Подобие или история эксплуатации:** прежний зачет верификации для идентичных или почти таких же систем.

Тестирование

Тестирование предполагает фактический запуск ПО и работу аппаратуры, возможно в целевой среде, представляющей реальную эксплуатационную авиационную систему. Основной акцент должен быть на требованиях, но тестирование также может учитывать фактическую структуру логики и ее реализацию. Например, для более критичной логики дополнительно оцениваются робастность логики и анализ структурного покрытия этой логики в ходе тестирования, основанного на требованиях. На ранних этапах развития тестирования авиационных систем считалось, что тестирование непосредственно повышает качество, и поэтому акцент делался на тестировании каждого компонента аппаратуры и ПО. А в отношении ПО считалось, что ПО есть лишь совокупность компонентов и потому тщательное тестирование каждого компонента улучшит качество ПО. Несостоятельность этих заблуждений быстро стала очевидной. По мере развития мы пришли к пониманию, что тестирование – это лишь воздействие на объект в заданной среде и последующее измерение результатов. В любой нетривиальной ситуации число задействованных переменных делает исчерпывающее тестирование невозможным. По своей природе тестирование является выборочной деятельностью, по результатам которой делают общие выводы. “Искусство” тестирования заключается в том, чтобы извлечь максимум информации из относительно малого набора тестов, который можно практически выполнить.

Тестирование не повышает качество само по себе. Оно дает важный способ оценить качество; при надлежащей обратной связи тестирование помогает понять и количественно оценить возможные недостатки, а затем устранить их через улучшение требований, проектных данных и реализации.

По мере усложнения программных продуктов тестирование эволюционировало.

Все чаще программные продукты представляют собой совокупности компонентов, взаимодействующих сложным и тонким образом. Но ПО является чем-то большим, чем просто совокупность компонентов, поскольку способ их взаимодействия связан с качеством системы. Поэтому по мере развития и совершенствования документов семейства DO-XXX в них тестирование уточнялось с акцентом на интеграционное и системное тестирование, а не на тестирование отдельных компонентов.

Для авиационного тестирования в соответствующем плане верификации следует рассмотреть следующую информацию (опять же через объективные критерии оценки):

Темы планирования верификации, обычно объединяемые с планированием валидации

- Цели и стратегии валидации и верификации (Validation and Verification, V&V) требований, проекта, кода или логики и самого тестирования.
- Взаимосвязь функционального, робастного и структурного тестирования, а также мест их выполнения.
- Контрольные перечни или ссылки на них для всех мероприятий валидации и верификации и разных видов V&V: моделирование, рассмотрение, тестирование, анализ, демонстрация, инспекция и т. д.
- Системные испытания, испытания аппаратуры и тестирование ПО, регрессионный анализ и регрессионное тестирование, независимость, инструменты и среда тестирования.
- Критерии перехода для валидации, верификации и трассируемости.

Авиационные и авионические системы проходят испытания на уровне системы обычно по ARP4754A, тогда как ПО внутри этих систем тестируется по DO-178C, а аппаратура проходит верификацию по DO-254 (Design Assurance Guidance for Airborne Electronic Hardware). Исключение составляют наземные и спутниковые системы, подпадающие под DO-278A, однако их аппаратура не проходит сертификацию вне системы, поэтому DO-254 не применяется; это связано с тем, что наземные системы широко используют готовые коммерческие изделия (Commercial Off-The-Shelf, COTS), в том числе аппаратуру, а испытания этих систем включают интегрированную аппаратуру. На уровне системы метод верификации обычно выбирают из следующих четырех ключевых методов или распределяют между ними:

1. Испытание
2. Демонстрация
3. Анализ
4. Инспекция

Хотя не существует общепринятой формулы, определяющей, какой метод следует применять и когда, либо относительный приоритет четырех указанных выше методов верификации, по мнению автора, относительный приоритет должен быть именно в приведенном выше порядке.

Метод = Испытание

Испытание предполагает фактическую работу проверяемого объекта, чтобы определить, соответствуют ли фактические результаты ожидаемым результатам, и является наиболее предпочтительным из четырех основных методов, поскольку оно менее субъективно, более повторяемо и наиболее близко воспроизводит реальные авиационные операции. Метод испытания кратко представлен на следующем рисунке:

Краткое описание метода испытания

Краткое описание метода испытания

- **Повторяемое объективное свидетельство:** работа системы и сравнение результатов.
- **Независимо разработано и оценено:** верификатор отличается от разработчика.
- Обычно перед началом испытаний проводится рассмотрение готовности к испытаниям (Test Readiness Review, TRR), где оценивают достаточность и проработанность запланированных испытаний.

Метод = Демонстрация

Демонстрация предполагает фактическую работу проверяемого объекта, при которой визуальные выходные данные (например, данные, видимые на индикаторе в кабине или на дисплее наземного диспетчера) сравниваются с ожидаемыми визуальными результатами. Для демонстрации характерно ручное или полуавтоматизированное проведение с участием оператора испытаний, который осуществляет визуальную оценку. Поскольку присутствует оператор или инспектор, крайне важно, чтобы визуальные критерии приемки были сформулированы очень четко для минимизации субъективности. Метод демонстрации кратко представлен на следующем рисунке:

Краткое описание метода демонстрации

Краткое описание метода демонстрации

- **Повторяемое объективное свидетельство:** работа системы и сравнение ожидаемых визуальных результатов с фактическими.
- **Независимо разработано и оценено:** верификатор отличается от разработчика.
- **Распространенная ошибка:** слабо сформулированные критерии приемки; необходимо обеспечить объективность.

Метод = Анализ

Анализ использует дополнительные научные или математические критерии, применяемые посредством определенного процесса, для оценки корректности артефакта. Анализ часто является вторичным сравнительным методом, применяемым тогда, когда демонстрация, испытание или инспекция недостаточны для оценки результатов. Примерами анализа являются просмотр матрицы двунаправленной трассируемости для выявления отсутствующих или некорректных связей, а также оценка внутренних элементов модели ПО или ветвей кода для определения, достигнут ли достаточный уровень покрытия. Метод анализа кратко представлен на следующем рисунке:

Краткое описание метода анализа

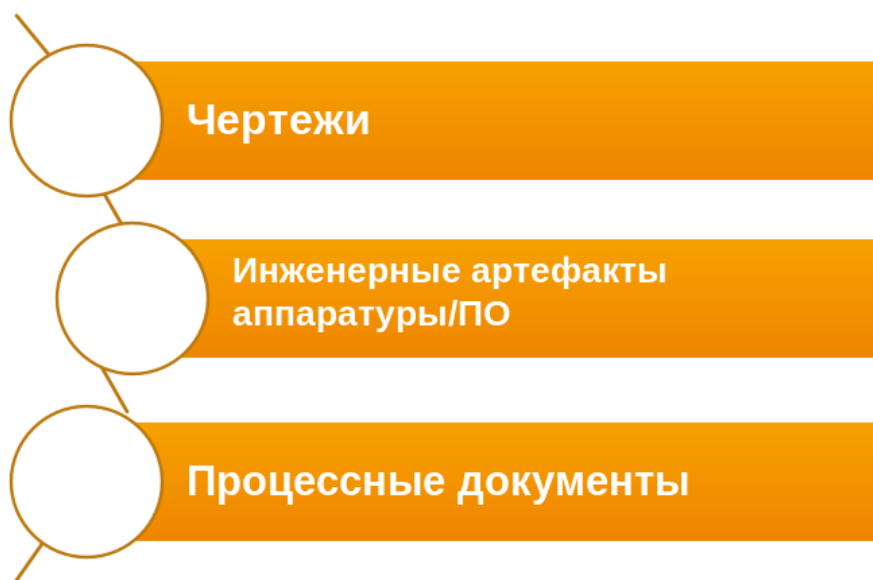
Краткое описание метода анализа

- Детальное изучение артефакта, часто с применением вторичного сравнения.
- Может включать моделирование и анализ покрытия.
- Следует учитывать нормальные и ненормальные условия эксплуатации.

Метод = Инспекция

Инспекция предполагает визуальное изучение артефакта, который не выполняется динамически и потому считается “статическим”. Артефакт изучается, чтобы определить, присутствуют ли определенные признаки (или чтобы подтвердить, что признаки отсутствуют). Часто перечень признаков формируется из характеристик, которые заказчики артефакта считают желательными или проблемными. Например, часть испытательного оборудования может быть проинспектирована, чтобы удостовериться в наличии USB-соединения или в том, что это соединение не разъединяется при воздействии нагрузки. Часто инспектируются планы и созданные инженерные артефакты, а также записи о калибровке. Область применения метода инспекции кратко представлена на следующем рисунке:

Краткое описание метода инспекции



Инспекции кода (их также называют рассмотрениями кода или пошаговыми разборами кода) представляют собой особую форму инспекции, которая технически относится к категории “рассмотрение”, описанной выше. В авиации тестирование формализуется через план, описывающий общий подход к тестированию, а также через тестовые процедуры и результаты, полученные после выполнения этих процедур.

Тестовые процедуры должны задавать следующую информацию или ссылаться на нее:

Содержание тестовых процедур

- Цель тестирования.
- Требования, охватываемые тестами или их частями.
- Среда тестирования и подготовка.
- Входные воздействия, действия оператора, последовательности и временные зависимости.
- Ожидаемые результаты и допуски, включая тесты на робастность.

Результаты тестирования должны задавать следующую информацию или ссылаться на нее:

Содержание результатов тестирования

- Данные идентификации системы и версии.
- Идентификатор тестовой процедуры, входные данные и версия.
- Идентификация инструментов и оборудования с указанием версии.
- Результаты тестирования, включая объективное решение "прошел/не прошел" и обоснование ожидаемых и фактических результатов.

Тестирование логики ПО и аппаратуры

В авиации существуют некоторые виды аппаратных изделий, которые считаются простой электронной аппаратурой (Simple Electronic Hardware, SEH) и потому не требуют специального тестирования; они лишь документируются, определяются, конфигурируются, контролируются и верифицируются в контексте системы, в которую они установлены. Например, резисторы, конденсаторы и автономные дискретные устройства, такие как элементы И/ИЛИ, не требуют дополнительного специализированного тестирования, а лишь простого подтверждения того, что определенные системные испытания активируют их и, следовательно, обнаружили бы дефекты. Для таких простых устройств, закупаемых как готовые коммерческие изделия (Commercial Off-The-Shelf, COTS), допускается презюмируемое соответствие уровню гарантии разработки А при условии их правильного выбора и управления, как описано выше. Однако аппаратная логика, реализованная в интегральной схеме, почти всегда является сложной, а не простой, потому что пользователь не может доказать, что оценил эту логику для всех “ожидаемых условий эксплуатации”. Помните: для всех асинхронных систем, кроме самых тривиальных, число ожидаемых условий эксплуатации стремится к бесконечности; следовательно, без формальной математической модели ошибочно считать, что можно доказуемо верифицировать бесконечное число вариантов входов и выходов во временной области. (Даже при наличии такой формальной математической

модели необходимо доказать математическую замкнутость по дополнению DO-333 по формальным методам (Formal Methods Supplement to DO-178C and DO-278A), чтобы допустимо сократить традиционные мероприятия тестирования, описанные здесь.) Поэтому авиационная логика требует определенного инженерного жизненного цикла, основанного на свидетельствах, чтобы компенсировать иначе бесконечное число ее тестовых примеров и доказать корректность. Современные системы, будь то аппаратура или ПО, объединяют простую аппаратуру или отдельные инструкции кода в более полезную, но и более сложную систему. Хотя мы можем предполагать, что отдельный конденсатор или инструкция сложения будут работать как ожидается, именно логическая структура определяет, можно ли на них полагаться. Эта логика тестируется на нескольких уровнях, как показано на следующем рисунке:

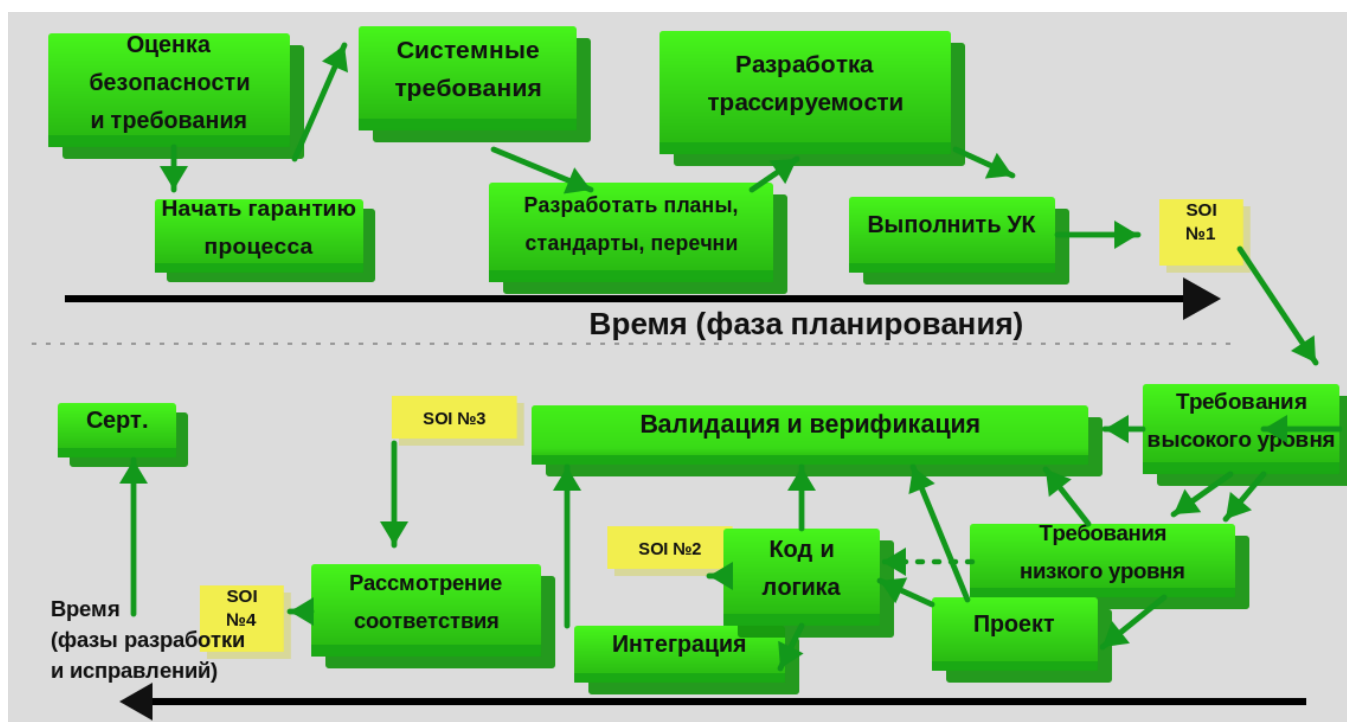


Типовой V-модельный подход к авиационному тестированию

Авиационные руководящие документы обычно основаны на “V-модели”, показанной выше, поскольку время разработки этих документов совпало с признанием V-модели. В V-модели сначала разрабатываются требования к воздушному судну, затем требования к системе, затем начальные требования к ПО и аппаратуре, затем детальные аспекты ПО и аппаратуры, то есть “сверху вниз”, что представляет левую сторону “V” на рисунке выше. Затем тестирование выполняется снизу вверх, начиная с логики аппаратуры и ПО, затем интеграции, затем системы и, наконец, воздушного судна.

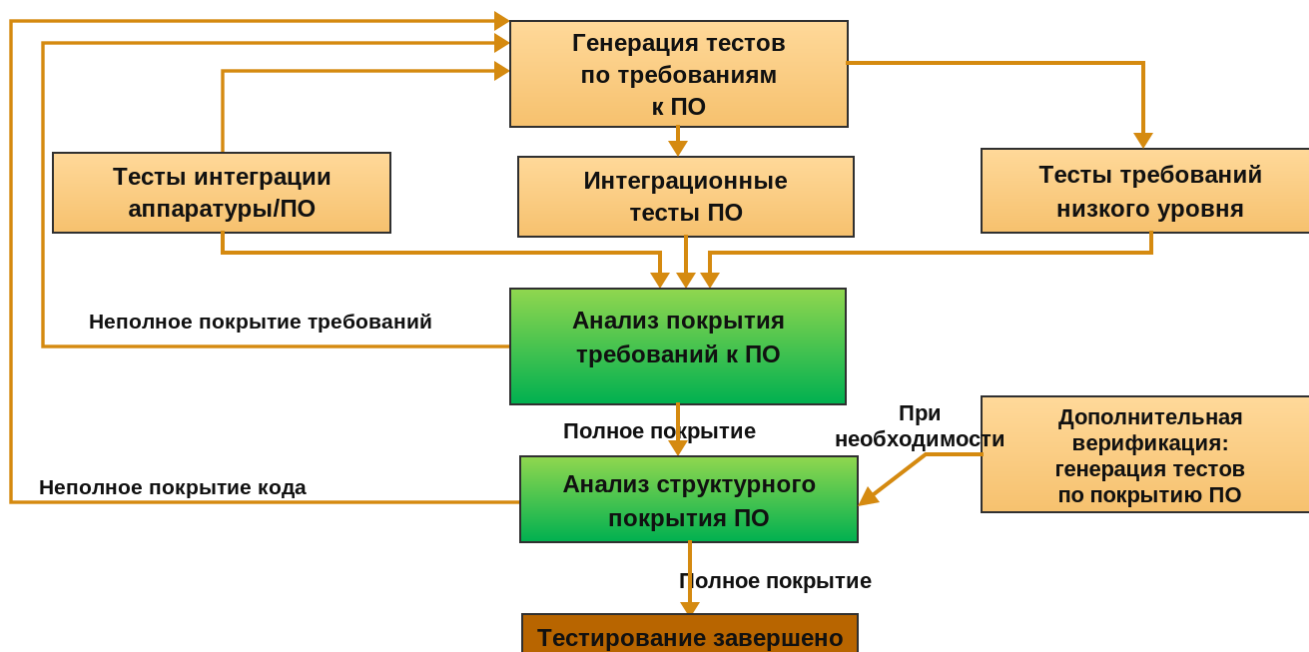
Для авиационных систем, содержащих логику, применяются дополнительные правила разработки и верификации, поскольку иначе нет способа обеспечить корректность этой логики. Когда логика выполняется в форме микропроцессора или микроконтроллера, она называется ПО (software), поскольку логические инструкции являются “мягкими”: сначала они компилируются из исходного кода, затем линкуются для построения исполняемого файла, затем загружаются в некоторую форму оперативного запоминающего устройства (Random Access Memory, RAM; ОЗУ) и, наконец, выполняются на готовом коммерческом процессоре, содержащем счетчик команд и регистры. Таким образом, ПО сертифицируется по DO-178C для бортовой авионики и по DO-278A для наземных и космических систем. Авиационное ПО чаще всего пишется на языке высокого уровня (High-Order Language, HOL), таком как C, C++ или Ada; Java все чаще используется для наземных систем и бортовых систем с меньшей критичностью с точки зрения безопасности (Java – хороший язык и широко используется в областях, не критически важных с точки зрения безопасности, однако существуют разные мнения о динамическом использовании памяти во время выполнения и о соответствующих стандартах безопасного кодирования для Java). Напротив, когда логика физически реализована в интегральной схеме, она называется сложной электронной аппаратурой (Complex Electronic Hardware, CEH) и, если она бортовая, сертифицируется по DO-254. Сложная электронная аппаратура обычно описывается на языке описания аппаратуры (Hardware Description Language, HDL), таком как Verilog или язык описания аппаратуры VHDL (VHSIC Hardware Description Language, VHDL).

Хотя все чаще предпринимаются попытки верифицировать корректность логики путем оценки семантики логики, авиация придерживается гораздо более консервативного и прагматичного взгляда, согласно которому для современных сложных систем это пока неосуществимо. Авиация требует упорядоченного и хорошо спланированного подхода к проектированию логики, а затем к ее верификации. Рисунок ниже суммирует мнение автора об оптимальном инженерном маршруте:



Именно этот последовательный поэтапный подход к разработке ПО, в котором каждое крупное мероприятие показано выше зеленым блоком, позволяет оценивать входные и выходные данные на всем протяжении жизненного цикла. Хотя для простоты валидация и верификация показаны как один зеленый блок, в действительности часть валидации и верификации, относящаяся к рассмотрениям, выполняется непрерывно и по выходным данным всех остальных этапов.

На рисунке ниже показан процесс тестирования и анализа авиационной логики. Обратите внимание: на рисунке используется термин “software”, но в этом контексте он может относиться и к ПО, и к аппаратной логике. Повторим: термин “исходный код” (“code”) обычно применяется к ПО, тогда как термин “логика” (“logic”) применяется к сложной электронной аппаратуре с аппаратной логикой, описанной, например, на VHDL. Однако процесс тестирования схож. Специалисты по тестированию бортовой авионической аппаратуры должны учитывать, что документ DO-254 фактически не охватывал такую аппаратную логику, поскольку DO-254 был инициирован в конце 1990-х годов, когда такая логика еще не была широко распространена в авионике. Как отмечено в главе 8 этой книги, DO-254 развивался через консультативный циркуляр (Advisory Circular, AC) 20-152, затем меморандум 27 Группы сертификационных органов по ПО (Certification Authorities Software Team, CAST-27), а затем пересмотренный международный документ приемлемых методов определения соответствия A(M)C 20-152A (Acceptable Means of Compliance, AMC), выпущенный летом 2020 года. Итог этих изменений DO-254 по существу состоит в том, что для целей сертификации к аппаратной логике, начиная с уровня гарантии разработки В, следует относиться почти так же, как к программному коду.



Реализация процесса тестирования и анализа авиационной логики

Приоритеты обнаружения ошибок зависят от уровня тестирования. Используя приведенный выше рисунок распределения тестирования, следующая таблица определяет приоритеты акцентов тестирования по уровням:

Приоритеты акцентов тестирования по уровням

Низкоуровневое тестирование	Интеграционное тестирование ПО	Тестирование интеграции аппаратуры и ПО
- Отказы алгоритмов - Некорректные операции циклов - Некорректные логические решения - Невыполнение обработки корректных комбинаций входных данных - Некорректная реакция на ошибочные входные данные - Некорректная обработка исключений - Некорректная последовательность вычислений - Недостаточные точность, правильность и производительность алгоритма	- Некорректная инициализация переменных - Ошибки передачи параметров (Глобальное) повреждение данных - Недостаточная численная разрешающая способность - Некорректная последовательность событий и операций	- Некорректная обработка прерываний - Временные характеристики и производительность - Переходные ошибки аппаратуры - Конкуренция за ресурсы - Ошибки обнаружения встроенным контролем (Built-In Test, BIT) - Некорректные контуры обратной связи - Некорректное управление устройствами - Переполнение стека - Некорректная проверка версии загрузки - Нарушения обособления ПО (Software Partitioning)

Как показано на двух предыдущих рисунках, процесс тестирования логики по существу можно разбить на следующие шаги с учетом обратных связей, показанных стрелками на рисунках выше:

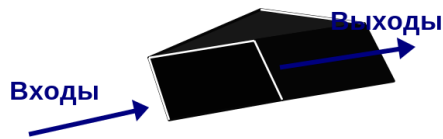
1. Начать с функционального тестирования (основанного на требованиях), назначая для отдельных требований один или несколько видов тестирования:
 - a. тестирование интеграции аппаратуры и ПО, и/или:
 - b. тестирование только ПО (или только аппаратуры в случае логики), и/или:
 - c. тестирование требований низкого уровня (с использованием только подмножества ПО или только аппаратуры в случае логики)
2. Выполнить это тестирование согласно шагу 1 выше и оценить полноту покрытия требований, включая покрытие как нормальных тестовых примеров, так и тестовых примеров для проверки робастности, описанных ранее в этой главе; если покрытие неполное, вернуться назад, добавить детализацию требований и повторить шаг 1.
3. Если для ПО уровень гарантии разработки C или выше, либо для сложной электронной аппаратуры уровень гарантии разработки B или выше, выполнить анализ структурного покрытия, требуемый для данного уровня гарантии разработки. Если обнаружены непокрытые структуры, удалить мертвый код, обосновать отключенный код (включая защитный код и код, используемый только в режиме технического обслуживания), а для любого прочего кода добавить требования, вернувшись назад и дополнив требования для покрытия таких непокрытых структур, после чего повторить шаг 1.

Черный ящик и белый ящик.

Строгость верификации логики соразмерна потенциальному риску этой логики, определяемому уровнем гарантии разработки элемента (Item Development Assurance Level, IDAL), связанным с этой логикой. Очевидно, что логика уровня гарантии разработки A оценивается строже, чем, например, логика уровня C. Поскольку логика уровня D для целей рассмотрений и тестирования считается “черным ящиком”, рассмотрение логики (аппаратной или программной) для уровня D не требуется. В конце концов, отказ ПО уровня D должен легко компенсироваться действиями экипажа или другими средствами авионики, поэтому он относится к категории “незначительный”, и затраты, связанные с рассмотрением логики и тестированием этой логики методом белого ящика, не требуются. Однако для ПО уровня C и выше, а также для логики уровня B и выше, дефекты могут иметь значительное эксплуатационное воздействие, поэтому к процессам проектирования и верификации применяется большая строгость. Эта дополнительная строгость называется верификацией методом “белого ящика”, что означает непосредственную оценку самой логики. На рисунке ниже показано различие между тестированием методом черного ящика и методом белого ящика:

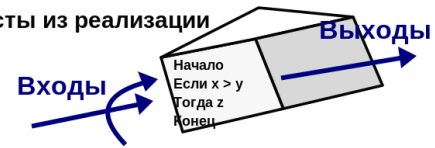
Черный ящик:

- Не учитывает реализацию
- Нельзя «видеть внутри ящика»
- Тестовые примеры из требований



Белый ящик:

- Зависит от реализации
- Нужно использовать реализацию
- Видно внутреннее устройство
- Тесты из реализации



Как показано выше, тестирование методом черного ящика означает, что вы не можете “видеть внутри ящика”, и поэтому тестирование основывается на связанных с проверяемым объектом требованиях. Напротив, тестирование методом белого ящика использует дополнительные оценки, основанные на непосредственном анализе самой исходной логики. На практике при оценке логики часто сочетают тестирование методами белого и черного ящика. Чтобы не допустить, чтобы специалист по тестированию начал оценивать логику до тестирования требований, связанных с этой логикой, лучше выполнять тестирование методом черного ящика до тестирования методом белого ящика; это делает тестирование в целом более объективным, поскольку специалиста по тестированию не вводит в заблуждение написание тестовых примеров по фактической логике вместо предпочтительного метода написания тестовых примеров по требованиям.

В авиации всегда требуется тестирование методом черного ящика, и существует единодушное мнение, что тестирование методом черного ящика эффективнее, чем тестирование только методом белого ящика, а также что тестирование методом черного ящика дает большую отдачу от инвестиций, чем метод белого ящика. Однако у тестирования методом черного ящика, по общему признанию, есть следующие недостатки:

- Они зависят от корректности и степени детализации требований.
- Они зависят от квалификации специалиста по тестированию в части выявления дефектов.
- С их помощью трудно полностью оценить атрибуты робастности, присущие требованиям (в целом), но некорректно или недостаточно обработанные в исходной логике.

Таким образом, авиационная логика в основном имеет четыре категории мероприятий по тестированию, как показано в следующем текстовом блоке:

Категории тестирования авиационной логики

1. **Функциональное тестирование:** проверяет все требования.
2. **Тестирование в нормальном диапазоне:** проверяет "штатные" условия.
3. **Тестирование на робастность:** проверяет "нештатные" условия.
4. **Анализ структурного покрытия:** подтверждает, что весь код получил требуемое покрытие.

Как показано выше, функциональное тестирование по сути является тестированием, основанным на требованиях, и относится к методу черного ящика, то есть тестовые примеры пишутся по требованиям, а не с учетом самой логики. В идеале это функциональное тестирование затем охватывает все нормальные условия эксплуатации и условия проверки робастности без необходимости изучения логики для такой оценки. Затем по ранее выполненному функциональному тестированию, тестированию в нормальном диапазоне и тестированию для проверки робастности выполняется анализ структурного покрытия, чтобы гарантировать достижение минимального покрытия логики, требуемого соответствующим уровнем гарантии разработки. И разумеется, именно так происходит в идеальном мире, но не в нашем реальном мире.

В реальном мире (в вашем мире) требования никогда не бывают идеально детализированы, а разработчики логики делают определенные допущения при проектировании и реализации своей логики. По этим причинам выполнение лишь функционального тестирования (основанного на требованиях) не позволит полностью оценить эксплуатационные условия, для обработки которых логика была разработана (или должна была быть разработана). Поэтому необходимо дополнительное тестирование методом белого ящика, которое выполняет непосредственную оценку логики после тестирования методом черного ящика. Например, после того как специалист по тестированию изучит логику и определит, было ли предыдущее функциональное тестирование достаточным, может потребоваться дополнительное тестирование в нормальном диапазоне и тестирование для проверки робастности. Если этот процесс также показывает, что необходимо добавить дополнительные требования, чтобы лучше обосновать такие дополнительные тестовые примеры, то эти требования добавляются, а для замыкания добавляется также тестирование, основанное на требованиях. Единственный способ объективно определить, была ли деятельность по тестированию минимально полной, – затем выполнить анализ структурного покрытия.

Тестирование на робастность

Тестирование на робастность – в определенной степени, к сожалению, субъективная область тестирования авиационных систем и логики. Оно отвечает на вопрос: “Достаточно ли робастен объект в части обработки нештатных условий?” Тестирование на робастность оценивает реакцию на:

Ненормальные условия эксплуатации; граничные значения; какие именно границы? На границе и за ее пределами! Битовая точность на границах? Ошибочные и недопустимые значения/переходы; стрессовое тестирование и тестирование производительности.

Причина некоторой субъективности тестирования на робастность состоит в том, что число условий робастности и их комбинаций обычно больше доступного времени и бюджета на тестирование, особенно для сложных систем, где это число бесконечно. Например, рассмотрим тестирование напряжения постоянного тока на воздушном судне, номинальное значение которого составляет 28 вольт, а допустимый диапазон напряжения – от 26,0 до 30,0 В. В таком случае было бы как минимум пять тестовых примеров:

Тестовый пример	Тестовое входное напряжение	Результат теста
1. Пониженное напряжение	25,99 В постоянного тока	Ошибка пониженного напряжения
2. В диапазоне, нижняя граница	26,0 В постоянного тока	В диапазоне, граница
3. В диапазоне, номинальное напряжение	28,0 В постоянного тока	В диапазоне, номинал
4. В диапазоне, верхняя граница	30,0 В постоянного тока	В диапазоне, граница
5. Повышенное напряжение	30,01 В постоянного тока	Ошибка повышенного напряжения

Разумеется, приведенный выше пример прост. Тестирование на робастность становится более сложным по мере учета многочисленных переходов состояний и комбинаций входных данных. Кроме того, тестирование производительности, стрессовое тестирование и анализ наихудшего времени выполнения (Worst-Case Execution Time, WCET) должны дополнительно определять ожидаемые наихудшие комбинации, максимально нагружающие логику.

Анализ структурного покрытия логики

Анализ структурного покрытия является завершающим мероприятием по оценке полноты тестирования применительно к логике. Как уже отмечалось, логика с более высокой критичностью требует тестирования методом белого ящика, причем объем такого тестирования возрастает вместе с уровнем критичности. Для бортового ПО уровня гарантии разработки С и выше, наземного и спутникового ПО уровня гарантии (Assurance Level, AL) 3 и выше, а также бортовой аппаратуры уровня гарантии разработки В и выше требуется тестирование методом белого ящика, включая анализ структурного покрытия.

ПО: уровень гарантии разработки С и уровень гарантии AL 3: Покрытие операторов

Покрывание операторов оценивает динамическое выполнение в ходе тестирования каждого соответствующего оператора исходного кода, где оператор обычно является наименьшей сущностью, полностью разбираемой компилятором, обычно обозначаемой разделителями в виде точки с запятой (“;”).

ПО и бортовая аппаратура: уровень гарантии разработки В и уровень гарантии AL 2: Покрывание операторов, условий и решений

Покрывание условий: каждое условие в каждом решении программы принимало все возможные исходы хотя бы один раз.

Покрывание решений: каждая точка входа и выхода в программе была пройдена хотя бы один раз, и каждое решение в программе принимало все возможные исходы хотя бы один раз.

ПО и бортовая аппаратура: уровень гарантии разработки А и уровень гарантии AL 1: Покрывание операторов, условий и решений, а также модифицированное покрытие условий/решений (Modified Condition/Decision Coverage, MC/DC)

Модифицированное покрытие условий/решений: каждое **решение** принимало все возможные исходы хотя бы один раз, и показано, что каждое **условие** в решении **независимо** влияет на исход этого решения, при этом условие считается независимо влияющим на исход, если изменение только этого условия меняет исход решения.

Анализ структурного покрытия выполняется по трем причинам:

1. **Причина 1:** обеспечить выполнение тестирования для покрытия кода пропорционально уровню гарантии разработки (критичности); и
2. **Причина 2:** убедиться, что мертвый код отсутствует, а отключенный код идентифицирован, надлежащим образом обработан и обоснован; и
3. **Причина 3:** оценить качество требований к ПО посредством функционального тестирования этих требований.

Задумайтесь над этими тремя причинами анализа структурного покрытия:

Причина 1 выше гарантирует, что более критичная логика (более высокий уровень гарантии разработки) имеет большее число своих потенциальных путей, покрытых формально трассируемыми тестами для каждого такого пути. Следовательно, более критичная логика имеет меньшее число возможных непроверенных эксплуатационных условий, которые могли бы возникнуть при реальной эксплуатации. Помните, что в сложных асинхронных системах, присущих авиации, число комбинаций входов/событий бесконечно; полностью протестировать бесконечный набор комбинаций входных данных невозможно. Анализ структурного покрытия, применяемый в авиации, – возможно, лучший способ по крайней мере гарантировать покрытие наиболее вероятных и очевидных комбинаций. (Некоторые передовые авиационные инженеры, включая автора,

полагают, что формальные методы, описанные в DO-333, а также искусственный интеллект когда-нибудь смогут уменьшить значение структурного покрытия. Но пока не сегодня.)

Причина 2 выше выполняет цели сертификации для тестирования методом белого ящика, обеспечивая, чтобы вся логика имела основание для существования (то есть двунаправленную трассируемость к формальному требованию).

Причина 3 выше многие специалисты по тестированию авиационной логики и гарантии качества считают самой важной из трех причин. Как отмечалось выше, анализ структурного покрытия является заключительным мероприятием тестирования и выполняется только после завершения всего предшествующего функционального тестирования, основанного на требованиях. Если соответствующие требования и их функциональное тестирование были разработаны в соответствии с авиационными руководящими документами и целями, непокрытых логических структур и путей должно остаться очень мало.

Причина 3 выше – очень важная идея: сертифицирующие органы, а иногда даже персонал гарантии качества, не знакомы детально с требованиями к изделию и не могут напрямую оценить техническое качество этих требований и связанного с ними тестирования. Но в конечном счете им нужен механизм для этого. Таким механизмом является причина 3: если требования к логике достаточно детализированы, а функциональное тестирование этих требований достаточно тщательно, анализ структурного покрытия должен легко показать, что 90-95 % логических структур и путей были покрыты функциональным тестированием. Вот и механизм. И наоборот, если функциональное тестирование покрыло 75 % или менее логических структур и путей, очевидно, что и детализация требований, и тщательность тестирования недостаточны; следует вернуться назад и добавить требования, чтобы достичь 95 %-ного покрытия, а затем повторять, пока 100 % не будут покрыты или обоснованы как отключенный код.

Теперь рассмотрим следующие определения, относящиеся к структурному покрытию:

Определения

Условие: логическое выражение, которое является неделимым (атомарным). Его также часто называют логической переменной: она может принимать только значения “Истина” или “Ложь” и не делится на более простые подкомпоненты.

Решение: логическое выражение, которое может состоять из нескольких условий, разделенных логическими операторами, такими как “ИЛИ”, “И” и т. п.

Модифицированное покрытие условий/решений: каждое решение принимало все возможные исходы хотя бы один раз, и показано, что каждое условие в решении независимо влияет на исход этого решения. (Условие независимо влияет на исход решения, если только это условие меняет исход.)

В качестве примера структурного покрытия рассмотрим следующий фрагмент кода:

```
if ( (A || B) && C )
{
    /* <Insert code instructions here> */
}
else
{
    /* <insert code instructions here> */
}
```

В этом примере A, B и C представляют логические выражения. Чтобы выполнить критерий покрытия условий для этого примера, A, B и C должны быть оценены как минимум по одному разу в “Истина” и по одному разу в “Ложь”, что обеспечивается следующими двумя тестовыми примерами:

1. A = True / B = True / C = True
2. A = False / B = False / C = False

Для покрытия решений выражение ((A or B) and C) также должно быть оценено хотя бы один раз как “Истина” и еще один раз как “Ложь”. Как отмечалось ранее:

1. A = True / B = True / C = True -> True
2. A = False / B = False / C = False -> False

Приведенное выше обеспечивает покрытие решений.

Теперь отметим, что модифицированное покрытие условий/решений требует, чтобы каждое логическое условие было оценено один раз как “Истина” и один раз как “Ложь”, и при этом влияло на исход решения. Это означает, что при переходе от одного тестового примера к другому изменение значения только одного условия должно менять исход решения. Но только по двум предыдущим тестовым примерам невозможно установить, какое именно условие влияет на оценку решения. Отсюда и необходимость модифицированного покрытия условий/решений (см. определение выше). Поэтому для решения с N условиями требуется минимум N+1 тестовых примеров, чтобы обеспечить модифицированное покрытие условий/решений. Поскольку в приведенном выше примере было три логических условия (A, B и C), достаточно следующих тестовых примеров:

1. A = False / B = False / C = True -> решение оценивается как “Ложь”
2. A = False / B = True / C = True -> решение оценивается как “Истина”
3. A = False / B = True / C = False -> решение оценивается как “Ложь”
4. A = True / B = False / C = True -> решение оценивается как “Истина”

Как показано выше на примере модифицированного покрытия условий/решений:

- между первым и четвертым тестовыми примерами только А изменило свое значение, что также изменило исход решения (с “Ложь” в первом случае на “Истина” во втором);
- аналогично, между первым и вторым тестовыми примерами только В изменило свое значение, что также изменило исход решения (с “Ложь” на “Истина”);
- между вторым и третьим тестовыми примерами только С изменило свое значение, и исход решения также изменился (с “Истина” на “Ложь”).

Отключенный код и мертвый код

Хотя аппаратура может считаться “простой” или “сложной” (подробности см. в главе 8 этой книги), ПО всегда считается по своей сути сложным. По существу, существует пять категорий исходного кода ПО (для простоты именуемого далее “кодом”), которые перечислены ниже:

5 категорий кода

1. **C1 – исполняемый исходный код.** Нормальный код.
2. **C2 – неисполняемый исходный код.** Отключенный код согласно DO-178.
3. **C3 – неисполняемый исходный код, исключаемый на этапе компиляции.** #IFDEF, умный линкер, ...
4. **C4 – резервный неисполняемый исходный код.** Неиспользуемые библиотеки, функции операционной системы реального времени, OCPB (Real-Time Operating System, RTOS), ...
5. **C5 – неисполняемый исходный код, вызванный ошибкой проектирования.** Мертвый код согласно DO-178.

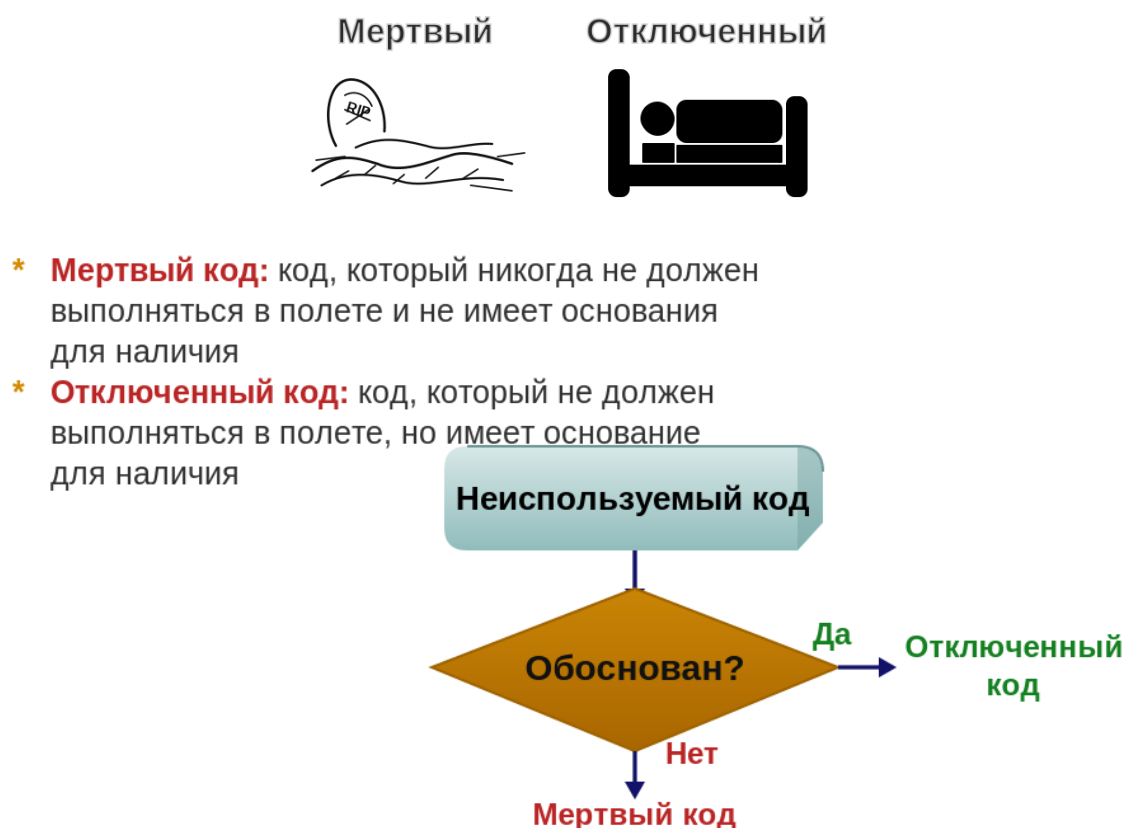
Считается, что все эксплуатационное авиационное ПО относится к одной из приведенных выше категорий. Для категорий 2 и 5, то есть соответственно отключенного кода и мертвого кода, по DO-178C и DO-278A должны применяться специальные правила. Сначала приведем полезные определения:

Мертвый код: *исполняемый объектный код (или данные), который вследствие ошибки проектирования не может быть выполнен (код) или использован (данные) в эксплуатационной конфигурации среды целевого вычислителя и не трассируется на системное требование или требование к ПО.*

Отключенный код: *исполняемый объектный код (или данные), который по замыслу проекта либо (а) не предназначен для выполнения (код) или использования (данные), например часть ранее разработанного программного компонента, либо (б) выполняется*

(код) или используется (данные) только в определенных конфигурациях среды целевого вычислителя, например код, включаемый выбором контакта в аппаратуре или программируемыми опциями ПО.

Для ПО уровня гарантии разработки С по DO-178С (или ПО уровня гарантии AL 3 по DO-278А) необходимо отдельно работать с мертвым и отключенным кодом. С мертвым кодом все просто: если он не трассируется на требование к ПО, а тестирование или анализ показывают, что он не может быть выполнен, он должен быть удален – без исключений. Однако отключенный код по существу является “спящим” кодом и требует дополнительной работы по проектированию и верификации. Сначала приведем простой рисунок, помогающий запомнить различие:



Во многих случаях наличие отключенного кода нормально и даже желательно. Следующий текстовый блок показывает четыре распространенные (и правильные) основные причины наличия отключенного кода:

Четыре основные причины наличия отключенного кода

1. ПО технического обслуживания.
2. Различные конфигурации воздушного судна.
3. Повторно используемое ПО с неиспользуемыми компонентами.
4. Защитный код.

Очевидно, что в указанных выше четырех случаях отключенный код желателен. Более того, большинство читателей к этому моменту книги уже должны понимать, что разработка авиационного ПО весьма дорога из-за строгости DO-178C и DO-278A; дополнительная информация о стоимости приведена в главе 23 этой книги. Однако одновременно читатели должны понимать, что большинство проектов авиационного ПО в значительной степени повторно используют ранее существовавшее ПО или предыдущие конфигурации. Если это ПО было хорошо спроектировано для повторного использования с неизменными модулями, подавляющее большинство требуемых мероприятий для неизменных модулей повторять не нужно. Исключение составляет тестирование, показывающее отсутствие непреднамеренных побочных эффектов от новой конфигурации или другого ПО, которое было изменено. Следовательно, компоненты ПО, предназначенные для повторного использования, фактически требуют намеренно отключенного кода. Для отключенного кода должны быть рассмотренные требования, трассируемость и соответствующий анализ кода, показывающий, что он действительно отключен и что его непреднамеренная активация не создаст угрозы безопасности полетов. Помните: все опытные программисты на C допускали ошибки с указателями, и такие ошибки могут непреднамеренно привести к активации любого кода, даже кода, обозначенного как “отключенный”. В текстовом блоке ниже обобщается управление отключенным кодом:

Обращение с отключенным кодом

1. Проектный механизм должен подтверждать, что отключенные функции или компоненты не оказывают отрицательного воздействия на активные функции или компоненты.
2. Нужно предоставить свидетельства, показывающие, что отключенный код выключен в тех средах, где его использование не предполагается.
3. Следует применять полный объем DO-178 в соответствии с уровнем критичности: требования, документы, рассмотрения и тестирование.

Где выполнять тестирование

В идеальном мире все авиационное тестирование проводилось бы на реальной эксплуатационной аппаратуре в завершенной конфигурации, идентичной той, которая летает или эксплуатируется на земле. И в идеальном мире нам не были бы нужны ремни безопасности и вторые пилоты. В реальном мире бюджеты и графики не всегда позволяют ждать конца проекта для выполнения формального тестирования. Кроме того, у логики есть множество условий проверки робастности, до которых трудно добраться на уровне системы. Часто тестирование авиационной логики выполняется на более низком уровне, чем сама система, например на инструментальных компьютерах либо на симуляторах или эмуляторах, применяемых при разработке системы. Однако если используется не только

завершающее тестирование на уровне системы, должно быть показано, что поведение логики будет эквивалентно ее поведению в эксплуатационной системе. Следующий текстовый блок помогает показать, когда может применяться каждый вариант:

Верификация аппаратуры по DO-254: где выполнять тестирование?

Варианты среды: целевая аппаратура, инструментальный компьютер, модель или эмулятор.

- **Если возможно влияние аппаратуры:** выполнять испытания на целевой аппаратуре. Это относится к пакету поддержки платы (Board Support Package, BSP), вводу/выводу (Input/Output, I/O), прерываниям, испытаниям асинхронных систем, временным характеристикам, памяти и встроенному контролю.
- **Если влияния аппаратуры нет:** выполнять тестирование целевого двоичного кода на инструментальном компьютере, модели или эмуляторе. Это применимо к алгоритмам, логическим решениям и обработке исключений.

Наземные и спутниковые системы связи, навигации, наблюдения и организации воздушного движения (Communication, Navigation, Surveillance / Air Traffic Management, CNS/ATM) (см. главу 11 этой книги) испытывают аппаратуру в составе интегрированных систем; отдельного разделения между испытаниями “системы” и “аппаратуры” нет. Однако, хотя бортовая аппаратура также подвергается интегрированным системным испытаниям, для самой аппаратуры существуют дополнительные мероприятия по верификации по DO-254. Эти дополнительные процессы верификации по DO-254 суммированы на рисунке ниже:

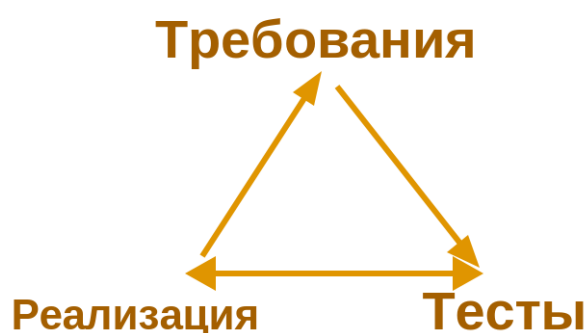
Дополнительные мероприятия по верификации аппаратуры по DO-254

Испытания	Рассмотрения	Анализы
– Испытания на основе требований к аппаратуре – Испытания логики (A/B)	– Требования – Концептуальный и детальный проект – Логика – Результаты испытаний на основе требований – Рассмотрение результатов испытаний низкого уровня (A/B)	– Анализ трассируемости (A/B/C) – Анализ покрытия требований (A/B/C) – Анализ покрытия испытаний низкого уровня (A/B) – Анализ структурного покрытия (A/B) – Общий анализ системы (A/B/C): – Статический временной анализ – Использование устройства – Тепловой режим/питание – Анализ покрытия выводов (A/B/C)

Временные характеристики бортовой аппаратуры необходимо верифицировать в рамках проекта. При этом следует учитывать температурные изменения и изменения питания, воздействующие на устройство, а также вариации технологического процесса

изготовления полупроводникового прибора, заданные его изготовителем. Статический временной анализ (Static Timing Analysis, STA) при наличии необходимых временных ограничений и условий является одним из возможных способов подтвердить соответствие этой цели для цифровой части заказных устройств.

Как объяснялось выше, тестирование авиационных систем, ПО и аппаратуры многогранно, и для сложных систем оно никогда не может быть доказуемо полным; число потенциальных тестовых примеров бесконечно, поскольку всегда можно добавить еще один тестовый пример. Со временем нередко на тестирование тратится больше денег и времени, чем потребовалось на исходную разработку. Поэтому автоматизация тестирования и эффективное распределение методов тестирования имеют первостепенное значение. Необходимо доказуемое покрытие тестами, а также двунаправленная трассируемость между требованиями, реализацией и тестами, как показано здесь:



В заключение отметим, что авиационное тестирование в действительности весьма отличается от тестирования в области информационных технологий (Information Technology, IT; ИТ), потребительских устройств и других областей, критически важных с точки зрения безопасности. Были написаны десятки отличных текстов, посвященных исключительно тестированию; эта краткая глава должна дать читателю инструменты, чтобы понять, чего требует авиация и чем испытания авиационных систем отличаются от испытаний в этих других областях.

Испытания на внешние воздействия по DO-160

Автор: Вэнс Хилдерман

Рецензенты:

- г-н Bill De Leeuw, уполномоченный представитель Федерального управления гражданской авиации США (Federal Aviation Administration, FAA) по системам, оборудованию и ПО
- г-н Alan Hour, независимый эксперт по DO-160

DO-160, **“Условия окружающей среды и методики испытаний бортового оборудования”** (*Environmental Conditions and Test Procedures for Airborne Equipment*), применяется практически ко всем коммерческим системам авионики и ко многим другим видам бортового оборудования. Уже само название DO-160 многое объясняет: этот документ:

- относится к испытаниям на внешние воздействия, а не к выполнению логики или процессам разработки;
- задает явные независимые критерии испытаний, которые нужно выполнить для сертификации оборудования;
- применяется к бортовому оборудованию и к ожидаемым наихудшим условиям внешней среды, которые потенциально могут возникнуть при эксплуатации воздушного судна.

По существу, DO-160 предписывает испытания, подтверждающие, что оборудование продолжит работать требуемым образом в наихудших условиях внешней среды, которые потенциально могут возникнуть на воздушном судне. Его цель – безопасность полетов; коммерческие соображения сами по себе для DO-160 не важны. Некоторые по-доброму называют DO-160 режимом испытаний “Shake And Bake” – “трясти и печь”, потому что ранние испытания по DO-160 строились вокруг воздействия на аппаратуру экстремальной вибрации и температуры. Но DO-160 всегда был больше, чем просто “трясти и печь”: последние редакции включают множество дополнительных испытаний, в том числе давление, солевой туман, воду, радиочастотные воздействия, магнитные эффекты, молнию и многие другие условия внешней среды.

Для наших европейских коллег DO-160 очень часто указывается как набор требуемых методов испытаний. В Европе документ ABD0100 компании Airbus часто служил применимым “стандартом проектирования”; ABD0100 многократно ссылается на DO-160. Важно, что ABD0100 дополнительно содержит требования и рекомендации по проектированию, тогда как DO-160 ограничен методами и уровнями испытаний. По

мнению автора, эти требования и рекомендации по проектированию весьма полезны и в целом необходимы; однако они также считаются в некоторой степени субъективными, поэтому сертифицирующие органы США обычно их не применяют. Примечание: документ ED-14 Европейской организации по оборудованию для гражданской авиации (European Organisation for Civil Aviation Equipment, EUROCAE) технически эквивалентен DO-160.

Прежде чем идти дальше, обдумайте небольшой тест...

1. **Верно/неверно: DO-160 применяется исключительно к электронной аппаратуре.**
2. **Верно/неверно: DO-160 можно использовать для оценки срока службы и средней наработки на отказ (Mean Time Between Failures, MTBF).**
3. **Верно/неверно: Испытания по DO-160 обычно выполняются одновременно с испытаниями характеристик и функциональности логики аппаратуры/ПО.**
4. **Верно/неверно: DO-160 в основном посвящен испытаниям на температуру и вибрацию.**
5. **Верно/неверно: DO-160 является статичным документом и редко обновляется.**
6. **Верно/неверно: Все испытания по DO-160 следует выполнять на одном и том же экземпляре оборудования.**

Знаете все ответы? Пара вопросов с подвохом, но дальнейший текст поможет.

У DO-160 долгая история. Первая версия была выпущена в 1975 году, но сам документ происходит от DO-138, датированного 1958 годом. Поэтому это один из старейших документов авиационной сертификации, которые все еще применяются, пусть и в последней редакции. Многие другие документы серии DO относятся к конкретным аспектам аппаратуры, ПО, систем и процессов, а DO-160 часто считают их “дедушкой”, поскольку почти все такие системы в итоге должны пройти испытания по DO-160.

По существу, DO-160 описывает испытания оборудования на внешние воздействия для соответствия минимальным эксплуатационным стандартам характеристик (Minimum Operational Performance Standards, MOPS). Испытания должны выполняться в сертифицированной лабораторной среде с использованием сертифицированного и калиброванного оборудования. Такая лабораторная среда делает испытания объективными, стандартизованными во всем мире и повторяемыми. Обычно их проводят испытательные центры, независимые от разработчика, хотя у крупных компаний могут быть собственные специализированные среды испытаний по DO-160. Несертифицированные лаборатории или оборудование полезны как “инженерные инструменты”: они повышают уверенность в конструкции и сокращают фактическое время испытаний в сертифицированной лаборатории. Успешное завершение программы испытаний по DO-160 – это принятый отчет об испытаниях, желательно, но не обязательно, со всеми результатами “пройдено”. Хороший отчет об испытаниях

подготовить не просто; в таком отчете указывают сертифицированные лаборатории и перечень испытательных образцов, включая калиброванное испытательное оборудование. Испытания по DO-160 выполняются по заранее подготовленной и принятой заказчиком процедуре испытаний, на которую ссылается план испытаний.

Поскольку подходящие помещения и оборудование часто ограничены, бронируются сильно заранее и стоят дорого, опытные поставщик и заказчик считают качество планирования испытательной базы показателем компетентности поставщика.

DO-160 не предназначен для расчета средней наработки на отказ (MTBF) или срока службы. Для этого применяются, например, военный стандарт США MIL-STD-217 (Military Standard 217) и/или приемлемый коммерческий вариант. DO-160 предназначен для задания минимального набора комплексных испытаний на внешние воздействия, которые должна пройти бортовая авиационная аппаратура, помимо всех прочих испытаний, требуемых для разработки и установки. Хотя DO-160 ссылается на широкий набор минимальных эксплуатационных стандартов характеристик, многие единицы оборудования не требуют всех испытаний, перечисленных в DO-160: часть из них просто неприменима. Как и для любого бортового оборудования, точный набор испытаний нужно согласовывать с применимым сертифицирующим органом. Испытания, которые “не требуются” или “выполняются анализом”, надлежащим образом отмечаются в документах, затем согласовываются и принимаются заказчиком.

Из всех документов Радиотехнической комиссии по авиации (Radio Technical Commission for Aeronautics, RTCA) именно DO-160, по-видимому, обновляется наиболее регулярно и часто. Анализ авиационных происшествий, производственные инспекции, аппаратные технологии и методики испытаний постоянно развиваются и пополняют совокупность знаний, необходимых для создания более качественной аппаратуры. Особенно это касается разделов DO-160 с 15 по 23, относящихся к электромагнитным помехам и электромагнитной совместимости (Electromagnetic Interference/Electromagnetic Compatibility, EMI/EMC): быстро меняющиеся спецификации и требования к испытаниям требуют предельной внимательности. В этих разделах необходимо тщательно анализировать спецификацию оборудования поставщика (Supplier Equipment Specification, SES, или аналогичный документ), которая всегда имеет приоритет над общими критериями DO-160. Сегодня даже потребительские изделия, не критически важные с точки зрения безопасности, обычно должны выдерживать суровые повседневные условия. Возьмем обычный современный мобильный телефон: его регулярно роняют, оставляют на раскаленной панели автомобиля, заливают кофе, а иногда он даже переживает несколько собачьих укусов. И все же от него ожидают, что он продолжит работать, и обычно так и происходит. Но бортовая авиационная аппаратура работает в куда более жесткой среде, даже если вероятность собачьих укусов там мала. DO-160 регулярно обновляется, чтобы отражать новейшее понимание потенциальных отказов авиационной техники и аппаратуры с учетом постоянно меняющейся практики производства и испытаний

аппаратуры. Ирония в том, что именно мобильный телефон, со всеми его 3G, 4G, Bluetooth и Wi-Fi, отлично показывает, почему требования так быстро меняются и ужесточаются.

RTCA/DO-160 пересматривался с учетом новых и усовершенствованных методов испытаний. Категории, сложившиеся со временем, отражают достаточно зрелое понимание тяжести внешних воздействий, степени их снижения, достижимой за счет конструкции установки, и робастности, которую нужно заложить в оборудование для работы в эксплуатационной среде.

DO-160 задает критерии минимальных стандартных условий испытаний на внешние воздействия по отдельным категориям среды. Цель этих испытаний – в контролируемой среде оценить, способна ли бортовая аппаратура функционировать в искусственно созданных вероятностных условиях среды, применимых к оборудованию на борту воздушного судна с учетом накопленного опыта DO-160. DO-160 отделен от стандартов характеристик оборудования, которые регулируют разработку и функциональность логики аппаратуры/ПО внутри данной системы. DO-160 может использоваться совместно с внешними стандартами, а также с конкретными изменениями содержания DO-160 в применимых разделах, например уровней и длительности испытаний.

Уровни испытаний, которые нужно применять, сформировались на основе множества воздушных судов и множества редакций DO-160. Они зависят от трех критериев: типа воздушного судна, физического места размещения и уровня гарантии разработки (Development Assurance Level, DAL). Для одного и того же физического места размещения уровень испытаний на внешние воздействия может быть разным при разных DAL.

Конкретные категории испытаний DO-160

В руководящих документах по авиационному ПО и аппаратуре, таких как DO-178 и DO-254, не требуется разрушать аппаратуру, чтобы верифицировать характеристики ПО или аппаратной логики. Более того, если испытание может повредить аппаратуру, например при проверке условия перенапряжения, уже одной такой возможности достаточно, чтобы указать “анализ” как метод испытаний и сохранить аппаратуру. В DO-160 все иначе.

DO-160 требует выполнения большого числа испытаний по широкому набору категорий внешней среды. Ниже перечислены основные категории испытаний, приведенные в DO-160:

Раздел DO-160	Категория испытаний
4	Температура
4	Высота
5	Изменение температуры
6	Влажность
7	Эксплуатационный удар и аварийная безопасность
8	Вибрация
9	Взрывоопасная атмосфера
10	Водонепроницаемость
11	Восприимчивость к воздействию жидкостей
12	Песок и пыль
13	Стойкость к грибку
14	Солевой туман
15	Магнитный эффект
16	Электропитание
17	Скачок напряжения
18	Восприимчивость к кондуктивным помехам звуковой частоты
19	Восприимчивость к наведенным сигналам
20	Восприимчивость к радиочастотному воздействию (излучаемому и кондуктивному)
21	Эмиссия радиочастотной энергии
22	Восприимчивость к переходным процессам, наведенным молнией
23	Прямое воздействие молнии
24	Обледенение
25	Электростатический разряд (Electrostatic Discharge, ESD)
26	Огонь, воспламеняемость

Категории испытаний DO-160

Почти 500 страниц и 26 категорий испытаний – этого достаточно, чтобы считать DO-160 подробным стандартом. Если вы участвуете в разработке бортового оборудования, стоит приобрести официальный экземпляр последней редакции DO-160 хотя бы для общего понимания. Из-за длительных сроков разработки воздушных судов и эффекта “унаследованной спецификации” нужно внимательно учитывать, что полностью или частично могут потребоваться более ранние редакции. Недавно RTCA вынесла разделы руководства пользователя в отдельный документ RTCA/DO-357 “Дополнение к руководству пользователя по DO-160G” (User Guide Supplement to DO-160G). Этот

документ объясняет обоснование требований, дает руководство по их применению, комментарии, возможные методы поиска причин проблем и уроки, полученные из лабораторного опыта.

Должна ли ваша компания самостоятельно выполнять испытания по DO-160? Если это крупная компания, у которой уже есть собственные ресурсы по DO-160, то да: таким специалистам стоит сохранять загрузку и актуальность опыта. В остальных случаях разумнее передать испытания по DO-160 на аутсорсинг одной из десятков официально одобренных лабораторий по всему миру, которые понимают испытания по DO-160 лучше вас. При этом вам все равно нужно хорошо разбираться в DO-160, чтобы обеспечить соблюдение критериев. Как и в любой разработке авиационной продукции, исправить проблему в десять раз дороже, чем предотвратить ее. Глубокое понимание DO-160 как раз и помогает предотвращать проблемы.

На этапе разработки спецификации, если вы в нем участвуете, знание критериев спецификации испытаний, уровня гарантии разработки, выбора метода испытаний и критериев “пройдено/не пройдено” может существенно повлиять на стоимость разработки изделия. Как показывает раздел 22 DO-160, посвященный восприимчивости к переходным процессам, наведенным молнией, требование “нарушение работы не допускается” вместо “нарушение работы с самовосстановлением” может резко усложнить конструкцию. И снова, даже если единственный вариант получить “решение на основе опыта” – пригласить консультанта, правильно выбранный краткосрочный консультант, пусть и дорогой, может сэкономить намного больше своей стоимости и избавить от кошмарного “цикла повторной квалификации”.

Последовательность испытаний

Различные категории испытаний DO-160, перечисленные выше, независимы. Иными словами, связанные испытания внутри каждой категории обычно выполняются независимо от других категорий. При 26 базовых категориях возможны миллионы сочетаний и последовательностей. Непрактично ожидать, что изготовители оборудования будут подвергать свои изделия всем возможным сочетаниям и последовательностям условий внешней среды. Например, обледенение, солевой туман и грибок могут немного влиять на восприимчивость к радиочастотному воздействию. Однако испытания DO-160 на радиочастотную восприимчивость выполняются в статической лабораторной среде без учета совместного влияния обледенения, солевого тумана и грибка, тогда как высокоускоренные ресурсные испытания (Highly Accelerated Life Test, HALT) предполагают одновременное воздействие температуры и вибрации. Порядок испытаний не полностью произволен; когда требуются совокупные испытания, нужно соблюдать следующие правила:

Последовательность испытаний DO-160

1. Испытания на стойкость к грибку должны предшествовать испытаниям в солевом тумане.
2. Испытания песком и пылью должны проводиться после испытаний на стойкость к грибку, в солевом тумане и на влажность.
3. Испытания во взрывоопасной атмосфере и на воспламеняемость должны проводиться последними.

Формальные испытания по DO-160, выполняемые для получения сертификационного зачета, то есть квалификации, должны проводиться на блоках с серьезно поставленным управлением конфигурацией. Это управление конфигурацией распространяется на выпущенные схемы, перечни деталей, сборочные чертежи и процедуры приемочных испытаний (Acceptance Test Procedures, ATP, или аналогичные процедуры). Блоки, выбранные для квалификационных испытаний по DO-160, должны отражать обычную серийную практику производства; специально отбирать экземпляры с характеристиками лучше типичных запрещено. Как минимум на этих блоках должны быть выполнены приемочные испытания в ходе производства, как обычно, и после завершения испытаний. Обычно приемочные испытания выполняют несколько раз в ходе квалификации по DO-160, чтобы при отказе, который способна выявить только эта процедура, можно было сузить причину до конкретного подмножества испытаний. Успешное прохождение приемочных испытаний после завершения формальных испытаний DO-160 является требованием.

Часто из-за длительности некоторых испытаний по DO-160, сложностей с планированием или достаточно “грязных” физических последствий отдельных испытаний, например солевого тумана, используют более одного блока. С другой стороны, испытательные блоки могут быть очень дорогими, особенно когда они нужны для срочных поставок заказчикам, поэтому их доступность ограничена. Все используемые блоки должны иметь одну и ту же формальную конфигурацию и один и тот же номер изделия, различаясь только серийным номером. Иначе придется проходить нетривиальный процесс обоснования того, что разные конфигурации ведут себя одинаково. Как предостережение: опытные заказчики и сертифицирующие органы будут с подозрением относиться к “слишком большому числу блоков для испытаний”, и не без оснований. Нормально, когда один блок используется для всех испытаний на электромагнитные помехи и электромагнитную совместимость, а один блок – для температурных испытаний; конечно, это может быть и один и тот же блок. Использование разных блоков для низкой и высокой температуры или разных блоков для эмиссии и восприимчивости обычно не принимается. Даже если все блоки имеют одинаковую конфигурацию, фактические физические детали внутри них неизбежно различаются, пусть даже только из-за безобидных на вид отличий в трассировке или длине проводов.

Поэтому известно, что обычные вариации от экземпляра к экземпляру могут давать заметно разные температурные характеристики или характеристики электромагнитных помех и совместимости.

Если изменения в конструкции или применяемых деталях считаются “крупными”, что вполне может случиться, может потребоваться, и часто действительно требуется, новый номер конечного изделия. Если это произошло во время квалификации, может потребоваться начать испытания заново либо провести повторную квалификацию частично или полностью. Квалификация по сходству – инженерная задача совсем не простая.

Физические блоки, использованные для формальной квалификации, должны храниться длительное время после завершения испытаний и могут считаться собственностью заказчика. Их нельзя изменять, а после испытаний их дальнейшее использование, если оно вообще допускается, сильно ограничено.

Допуски DO-160 и испытательное оборудование

При проведении формальных испытаний по DO-160 должны использоваться обычные, принятые и установившиеся для DO-160 процедуры и оборудование; сертификация и калибровка требуют значительного опыта и ресурсов. Даже среди экспертов по DO-160 эту работу оставляют экспертам и специализированным службам. Спецификации и допуски сложны и конкретны. Испытательное оборудование, используемое при испытаниях по DO-160, должно иметь актуальное и действительное подтверждение соответствия национальным или международным стандартам; кроме того, должны быть указаны изготовитель, серийный номер, дата последней калибровки, дата окончания действия калибровки или период ее действительности. Нужно вести определенные виды записей о калибровке, в чем могут участвовать как внутренние, так и внешние организации. Следует отметить, что слово “калибровка” здесь включает измерение и проверку характеристик: часто используемое оборудование вообще не имеет механизма “калибровки”, оно либо “в спецификации”, либо подлежит замене или ремонту. Вся эта информация должна входить в испытательную документацию и, разумеется, в отчет об испытаниях по DO-160, который часто является частью отчета о квалификационных испытаниях (Qualification Test Report, QTR). По этим причинам испытания часто передают на аутсорсинг одной из многочисленных независимых организаций, выполняющих испытания по DO-160. От хорошего автора отчета о квалификационных испытаниях ожидается, что он подготовит требуемый отчет так, чтобы заказчик его принял, хотя обычно все равно возникают вопросы и нужны доработки. Неопытный автор такого отчета может создать проблемы, сопоставимые с проблемами неопытного руководителя испытаний или неопытного автора процедуры испытаний. Наставничество здесь имеет первостепенное значение.

Условия окружающей среды по DO-160

Для различных испытаний DO-160 требует проводить испытания в так называемых “приемлемых условиях окружающей среды”. Очевидно, что некоторые испытания сами изменяют эти условия, например температурные испытания, испытания на обледенение или высотные испытания. Но если условия испытания не требуют изменения температуры, влажности и давления, для испытаний по DO-160 должны поддерживаться следующие условия:

- Относительная влажность должна быть меньше или равна 85%.
- Температура должна находиться в пределах 15-35 градусов Цельсия.
- Давление должно соответствовать стандартному давлению для высоты от -1500 до +5000 футов, то есть 84-107 кПа.

Любопытно, что слишком многие испытательные лаборатории не имеют на виду сертифицированных и калиброванных средств измерения условий окружающей среды и не используют их во время испытаний.

Примечания по DO-160

Испытания по DO-160, при всей их широте, выполняются по одному виду внешнего воздействия за раз. Однако в реальном мире, хотя бы из-за давления сроков, нередко требуется выполнять несколько испытаний одновременно. К счастью, DO-160 создавался и поддерживается с учетом этого обстоятельства. Чтобы пояснить эту очень желательную, но крайне трудную задачу, стоит помнить: уровни и длительности, задаваемые отдельным испытанием DO-160, вероятно, значительно превышают уровень одного аналогичного фактора, реально возникающего на воздушном судне. Синергию внешних воздействий в реальном транспортном средстве, во множестве транспортных средств и в огромном числе конструкций практически невозможно точно специфицировать. Это важно понимать. Также полезно знать, что для одной и той же среды пределы на эмиссию или генерацию обычно намного ниже пределов восприимчивости. И здесь причина та же: на воздушном судне действует синергия множества источников, а требованиям нужен статистический, то есть вероятностный, запас. Например, предельные уровни кондуктивной эмиссии электромагнитных помех, вероятно, намного ниже уровней кондуктивной восприимчивости к электромагнитным помехам.

Итог по уровням и пределам:

- Применимые уровни DO-160 определяются типом воздушного судна, местом размещения блока или блоков, уровнем гарантии разработки и вероятностным пониманием условий; при этом один блок может быть соединен с другими блоками, расположенными в разных местах.
- Испытания выполняются по одному виду внешнего воздействия за раз; комбинированные эффекты реальных условий учитываются в DO-160 через уровень и длительность.

- Для одной и той же испытательной среды допускаемые пределы эмиссии намного ниже, чем требуемый предел невосприимчивости; доводы о повышении допускаемых пределов эмиссии ради прохождения испытаний на этом основании принимаются редко. В принципе это относится и к вибрации, и к удару.
- Многие спецификации или показатели измерения испытаний задаются в дБ (децибелах): изменение мощности на 3 дБ соответствует коэффициенту 2.

Несколько разделов DO-160 содержат специальные указания по выявлению потенциальных слабых мест конструкции, восприимчивости и передачи вредных эмиссий. Может потребоваться анализ чувствительных частот, подлежащих испытанию, включая механический резонанс, а также испытания в нескольких режимах работы для блоков, у которых таких режимов несколько. Поскольку некоторые режимы работы длятся очень недолго по сравнению со временем испытания, например запуск, используются установившиеся режимы. Если анализ не выявляет специальных частот, представляющих интерес, применяются значения по умолчанию.

По сравнению с DO-178 или DO-254 у DO-160 меньше субъективных ловушек.

В DO-160 есть, или по крайней мере предпринимается попытка установить, четкие критерии “пройдено/не пройдено”. Они должны быть заданы в процедуре испытаний и быть воспроизводимыми. Отказ при испытании должен быть рассмотрен и устранен. Процесс устранения может оказаться запутанным, но не обязан быть таким. Методы устранения или коррекции отказов при испытаниях включают повторное успешное испытание с обоснованием, описание аномалий и отклонений испытаний, замену испытательного оборудования с обоснованием, запрос на смягчение требований спецификации, изменение конструкции и повторную квалификацию.

DO-160 применяется к большей части бортового оборудования, но нужно учитывать тип воздушного судна, для которого это оборудование предназначено. Например, у винтокрылых летательных аппаратов вибрационная и ударная среда жестче, чем у коммерческих самолетов с неподвижным крылом; поэтому соответствующие критерии испытаний DO-160 для вертолетов строже. Кроме того, недавно предпринимались попытки гармонизировать некоторые испытания DO-160 с релевантными военными стандартами, такими как MIL-STD-810 (Military Standard 810). В целом желательно использовать аналогичные стандарты или ссылаться на них, если пользователь докажет, что такие стандарты удовлетворяют пороговым критериям DO-160 либо превосходят их.

Авионика непрерывно развивается, и DO-160 не может обновляться в реальном времени, чтобы немедленно учитывать все изменения. Для конкретных типов оборудования следует обращаться к техническим стандартным приказам (Technical Standard Orders, TSO). Каждые несколько лет RTCA обобщает предлагаемые изменения DO-160, обычно связанные с новыми технологиями, и публикует новую редакцию. Например, развитие глобальной системы позиционирования (Global Positioning System, GPS), глобальной навигационной спутниковой системы, ГНСС (Global Navigation Satellite System, GNSS), а также появление новых растворителей и эксплуатационных жидкостей

привели к необходимости пересмотра, так же как авиационный полнодуплексный коммутируемый Ethernet (Avionics Full-Duplex Switched Ethernet, AFDX), мобильные телефоны и потребительская электроника на борту будущих воздушных судов будут стимулировать дальнейшие обновления в ближайшей перспективе. Кроме того, текущие расследования авиационных происшествий могут давать сопутствующие результаты анализа, которые потребуют улучшенных испытаний, предположительно через DO-160.

Гарантия качества и гарантия процесса в авиации

Автор: Вэнс Хилдерман

Рецензенты:

- Дон Коулман, представитель Федерального управления гражданской авиации США по инженерной экспертизе (FAA DER, Federal Aviation Administration Designated Engineering Representative)
- Гамзе Эроглу, ведущий инженер по сертификации
- Джордж Мейер, представитель FAA по инженерной экспертизе (DER)

Что такое гарантия качества и гарантия процесса в авиации и зачем они нужны

Гарантия качества (Quality Assurance, QA), вероятно, является одним из самых важных аспектов сертификации авиационного ПО и аппаратуры. Для воздушного судна, систем и аппаратуры эту деятельность обычно называют гарантией процесса (Process Assurance, PA), а для ПО – гарантией качества, но значение гарантии качества и гарантии процесса от этого не меняется. При этом эта деятельность редко получает столько внимания и признания, сколько заслуживает ее роль. Более того, авиационная гарантия качества заметно отличается от гарантии качества в обычной промышленной разработке и в разработке потребительских изделий. Рассмотрите следующие утверждения и определите, истинны они или ложны. Ответы и пояснения приведены далее.

1. **И / Л:** Самая важная роль гарантии качества состоит в оценке качества конечного продукта.
2. **И / Л:** Персонал гарантии качества выполняет технические рассмотрения.
3. **И / Л:** Персонал гарантии качества оценивает соблюдение инженерами-разработчиками авионики критериев, заданных в требуемой версии DO-178C, DO-254 и/или DO-278A.
4. **И / Л:** Гарантия качества играет ключевую роль в рассмотрении требований, проекта и кода, а также проводит тестирование.
5. **И / Л:** Четыре этапа взаимодействия с сертифицирующим органом (Stage of Involvement, SOI) представляют собой четыре аудита гарантии качества в процессе разработки авионики.
6. **И / Л:** Для каждого проекта разработки авионики уровня А или В нужны как минимум три разных человека, и самый важный из них для сертификации – независимый специалист по гарантии качества.

Если приведенные выше вопросы действительно показались простыми, можно поздравить себя с настоящим знанием гарантии качества. Если же они только выглядели простыми, дальнейший материал как раз для этого случая.

Отвечать на эти вопросы без понимания общей роли гарантии качества в авиационной разработке – примерно как разбираться в преобразованиях Фурье, не зная математического анализа: простым смертным это недоступно...

При разработке воздушного судна, систем, аппаратуры и ПО гарантия качества и гарантия процесса, далее для краткости называемые просто гарантией качества, имеют три основные обязанности:

1. Обеспечить, чтобы инженерные планы и стандарты конкретного проекта соответствовали DO-178C/254/278A и ARP4754A.
2. Оценить, а затем обеспечивать, чтобы независимая инженерная организация следовала этим планам от начала проекта до завершения поставки.
3. Сформировать и сохранить свидетельства выполнения первых двух обязанностей в виде записей аудитов, с закрытием всех выявленных проблем, дефектов и улучшений процесса.

В других, неавиационных видах разработки “гарантия качества” часто означает более вспомогательную и пассивную измерительную роль. Например, Wikipedia уместно описывает гарантию качества как “систематическое измерение, сравнение со стандартом, мониторинг процессов и связанный с ними контур обратной связи, обеспечивающий предотвращение ошибок”. В большинстве отраслей служба гарантии качества подчиняется инженерной организации, помогает с техническими рассмотрениями документации и выполняет различные системные испытания или тестирование ПО ближе к концу жизненного цикла разработки. Но в авиации это не роли гарантии качества. Почему авиационная гарантия качества не выполняет эти привычные для промышленности и потребительского сектора функции? Если перенести такое традиционное, хотя и распространенное понимание гарантии качества в авиацию, сразу возникают слабые места:

- Кто обеспечивает правильность планов и стандартов проекта и их соответствие ARP4754A, DO-178C, DO-254, DO-278A и т. д.?
- Кто обеспечивает, чтобы применимые процессы были детерминированными, повторяемыми, четко определенными и соответствовали ARP4754A и документам серии DO?
- Кто обеспечивает, чтобы контур обратной связи инженерного процесса был корректно определен, а затем соблюдался?
- Кто отвечает за наличие доказательств того, что ошибки были надлежащим образом выявлены, классифицированы и устранены?

В авиационной разработке ПО, например по DO-178C и DO-278A, см. предыдущие главы, ответ на эти вопросы прост: гарантия качества. Она строится на том, что автор называет “пирамидой гарантии качества”, как показано ниже. Обратите внимание на базовые элементы и на задачи гарантии качества, которые надстраиваются над этим основанием.

Пирамида основных обязанностей гарантии качества ПО

1. Обеспечить, чтобы планы, стандарты и контрольные перечни проекта соответствовали отраслевым стандартам.
2. Оценить соблюдение инженерами планов, стандартов и контрольных перечней.
3. Сохранять записи по первым двум видам деятельности.

На уровне воздушного судна, системы и аппаратуры гарантия качества называется гарантией процесса, поскольку дополнительно выполняются аудиты поставщиков и оценки производства. Поэтому ниже показана эквивалентная пирамида гарантии процесса.

Пирамида основных обязанностей гарантии процесса для систем и аппаратуры

Снизу вверх:

1. Обеспечение соответствия планов, стандартов и контрольных перечней проекта DO-254 и рекомендуемым аэрокосмическим практикам серии ARP47XX (Aerospace Recommended Practice, ARP).
2. Оценка соблюдения инженерами планов, стандартов и контрольных перечней.
3. Аудит перехода к производству.
4. Аудит поставщиков.
5. Ведение записей аудитов и метрик.

ARP4754A и документы серии DO в некоторой степени гибки в том, как определяются, планируются и выполняются процессы гарантии качества. Но при всей этой гибкости авиация требует методично применять определенные и измеримые процессы, чтобы обеспечить качество продукции. Именно обеспечить, а не просто немного улучшить.

Довести слабый продукт до уровня посредственного потребительского изделия в авиации явно недостаточно. Качество авиационной разработки зависит от качества компонентов, из которых складывается сама разработка. Какие компоненты процесса авиационной разработки влияют на качество? Все. Одинаково ли они важны? И нет, и да. Нет, потому что их потенциальное влияние на качество различается. Да, потому что каждый компонент МОЖЕТ повлиять на качество, а значит каждый из них остается потенциальной точкой отказа в разработке. Поэтому авиационная гарантия качества применяет отраслевые руководства и рассматривает все аспекты процесса разработки авионики, чтобы определить, как независимо оценивать входные и выходные данные

этого процесса. Экосистема разработки авионики начинается с безопасности полетов и систем, а аппаратура и ПО затем рассматриваются как подсистемы. Следовательно, гарантия качества должна видеть всю экосистему целиком и одновременно разбираться с самыми мелкими инженерными процессами, которые могут повлиять на качество продукта. Начинается все с планирования гарантии качества...

План гарантии качества (для DO-254 и ARP4754A – план гарантии процесса), который обычно готовится службой гарантии качества и всегда подписывается ею, является одним из пяти проектных планов по документам серии DO. Для ARP4754A таких планов восемь, см. соответствующие предыдущие главы. Эти планы определяют, как проект намерен выполнить применимые цели документов серии DO. Но гарантия качества должна подключаться рано, еще на этапах определения безопасности полетов и системы, чтобы убедиться, что эти этапы соответствуют применимым процессам и что созданы надлежащие артефакты для последующей разработки ПО и аппаратуры. Авиационные руководства допускают гибкость процессов, но не являются мягкими рекомендациями “для галочки”. гарантия качества должна обеспечить выполнение следующих целей:

Ключевые мероприятия по гарантии процесса/качества

1. Планы и стандарты разработки подготовлены по рекомендуемым аэрокосмическим практикам ARP4754A/ARP4761A (Aerospace Recommended Practice, ARP), DO-178C, DO-254, DO-278A и другим применимым руководствам.
2. Планы и стандарты разработки соблюдаются всем связанным персоналом, включая поставщиков.
3. Критерии перехода удовлетворены.
4. Рассмотрения соответствия выполнены.
5. Аудиты выполнены, чтобы обеспечить наличие данных, подтверждающих изложенное выше.

Ключевые мероприятия по гарантии процесса и гарантии качества

На первый взгляд перечень целей авиационной гарантии качества кажется простым, почти очевидным. В других, неавиационных средах разработки “гарантия качества” действительно выглядит похожей: оценить реализацию продукта, чтобы измерить и повысить качество. Поэтому практически каждое изделие потребительской электроники сегодня проходит какую-либо базовую форму контроля или гарантии качества.

Но более внимательное рассмотрение авиационной схемы гарантии качества показывает, что ее руководство активнее и жестче. В авионике для каждой связанной с безопасностью полетов бортовой программной системы требуются пять планов. Соответствующие пять планов требуются и для заказной сложной аппаратуры с аппаратной логикой, реализованной в интегральных схемах. Все пять планов важны, а

Федеральное управление гражданской авиации США и Агентство Европейского союза по авиационной безопасности (European Union Aviation Safety Agency, EASA) разумно не указывают, “какой именно” план или какие цели важнее. Тем не менее эксперты по сертификации авионики в целом согласны, что самым важным является План сертификации, а остальные четыре обязательных плана обычно идут в таком порядке:

1. *План сертификации*
2. *План гарантии качества*
3. *План управления конфигурацией*
4. *План верификации (и валидации для аппаратуры)*
5. *План разработки*

Как видно из этого перечня, второй по важности план – План гарантии качества. Он должен описывать процессы гарантии качества, которые обеспечивают, чтобы каждая система:

- имела полный комплект планов и стандартов, охватывающих 100 % всех применимых целей документов серии DO;
- определяла процессы мониторинга разработки в соответствии с планами и связанными стандартами, включая критерии перехода, задающие входные и выходные признаки для каждого инженерного действия;
- имела механизмы оценки того, соответствуют ли фактически применяемые в ходе разработки процессы этим планам;
- имела процесс обратной связи и управления, при котором корректирующие действия сопровождают соответствующие дефекты, а записи подтверждают статус и соответствие.

В авиации каждая из ключевых областей разработки воздушного судна и систем имеет собственный план гарантии качества:

Воздушное судно – *План гарантии процесса воздушного судна*

Авионические системы – *План гарантии процесса системы*

ПО авионики, а также связи, навигации, наблюдения и организации воздушного движения (Communication, Navigation, Surveillance / Air Traffic Management, CNS/ATM) – *План гарантии качества ПО*

Аппаратура авионики – *План процесса гарантии аппаратуры*

Хотя это разные документы с разной областью гарантии и разными свидетельствами, эти планы на практике очень похожи и охватывают ключевые темы, показанные на следующем рисунке.

Ключевые темы плана гарантии качества

- Описать процессы гарантии качества проекта по документам серии DO и ARP4754A; цели включают одобрения, аудиты и независимость.
- Описать участие гарантии качества в планировании, этапах взаимодействия с сертифицирующим органом (Stages of Involvement, SOI) и сертификации.
- Подтвердить независимость гарантии качества, а также глубину и периодичность аудитов.
- Описать оценку гарантией качества переходов и рассмотрений соответствия.
- Описать ведение записей гарантии качества, контроль дефектов и улучшения процесса.

Роль гарантии качества

Многие компании, особенно традиционные компании с историей надежной разработки, умеют определить систему качества и следовать ей. Однако документы серии DO делают акцент, точнее возлагают бремя доказательства, на подразделение гарантии качества. Причина одновременно техническая и политически прагматичная: когда несколько людей и подразделений поровну отвечают за доказательство качества, легко получить взаимные обвинения и отсутствие настоящей ответственности. Когда отвечает одно подразделение, полномочия обозначены явно: независимая гарантия качества. Если в неавиационных отраслях гарантия качества участвует в технических рассмотрениях, выполняет тестирование и подчиняется инженерному руководству, то авиационная гарантия качества устроена заметно иначе.

На следующем рисунке показаны явные различия между инженерной деятельностью и гарантией качества в авиации.

Авиационная инженерия в сравнении с гарантией качества

Инженеры:

- создают инженерные артефакты;
- верифицируют их через рассмотрения, тестирование и анализ.

Гарантия качества (Quality Assurance, QA):

- утверждает планы и стандарты;
- аудирует работу инженеров;
- управляет записями аудитов.

Значит ли это, что гарантия качества единолично отвечает за качество, а все инженеры проекта могут расслабиться и не следовать планам и стандартам? Конечно нет. Инженерная организация обязана следовать всем применимым планам и стандартам, а это оценивается техническими рассмотрениями и аудитами гарантии качества. Отвечает ли

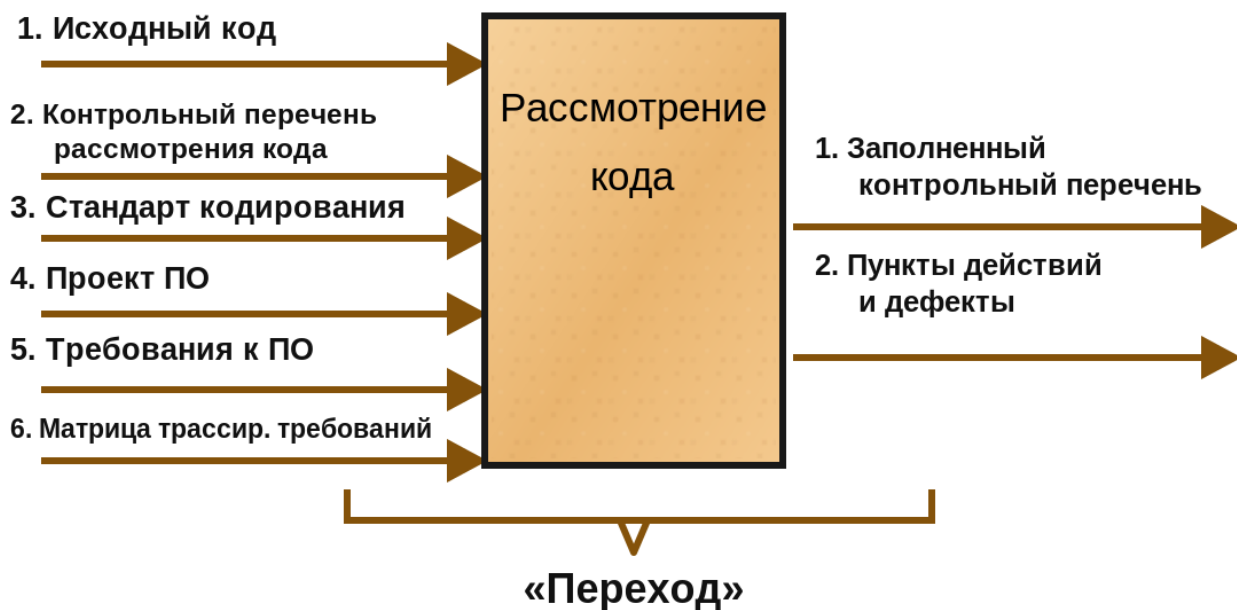
гарантия качества за оценку технического соответствия? Нет. Техническое соответствие оценивается рассмотрениями, а рассмотрения выполняет инженерная организация. Гарантия качества не выполняет технические рассмотрения. Вместо этого Гарантия качества проводит аудиты и оценивает соблюдение инженерной организацией установленного процесса. Не похоже ли это на словесное буквоедство, когда мы различаем рассмотрения и аудиты? В каком-то смысле да: гарантия качества “рассматривает” соблюдение инженерного процесса через аудиты. Но в документах серии DO это не называется рассмотрением, потому что рассмотрения там являются детальными техническими мероприятиями и выполняются инженерной организацией. Инженерная организация выполняет верификацию, а она включает рассмотрения, тестирование и анализ. Поэтому именно инженерная организация отвечает за рассмотрение инженерных артефактов, включая требования, проект, код и тестирование. Гарантия качества проводит аудиты, чтобы убедиться, что инженерная организация следует процессам рассмотрения, заданным в Плане верификации.

Аудиты гарантии качества

Аудиты гарантии качества не выполняются для каждого требования, проекта, элемента кода и теста, как это делается при инженерных рассмотрениях. Это одно из распространенных различий между документами серии DO и другими стандартами, где гарантии качества отводят активную, а иногда и ведущую, роль в технических рассмотрениях. По определению инженерное рассмотрение полностью применяется к заранее заданному контрольному перечню для всех требований и тестов, а при более высоких уровнях гарантии разработки (Development Assurance Level, DAL) – также для проекта и кода. В свою очередь аудиты гарантии качества являются выборочной проверкой каждого инженерного процесса и прежде всего охватывают четыре вида инженерной деятельности:

1. процессы разработки;
2. процессы управления конфигурацией;
3. процессы верификации;
4. инженерные рассмотрения, связанные с процессами разработки и верификации.

Итак, аудиты гарантии качества ориентированы на процесс, а инженерные рассмотрения – на техническое содержание. Рассмотрения, выполняемые инженерной организацией, и аудиты, выполняемые независимой гарантией качества, проводятся по заранее заданному шаблону контрольного перечня. Записи о соответствующем рассмотрении или аудите сохраняются. Гарантия качества также должна проводить аудит критериев перехода для каждого инженерного действия, то есть входных и выходных критериев, как показано на следующей схеме.



Авиационный пример перехода в разработке ПО: рассмотрение кода, аудит которого выполняет гарантия качества

Записи аудитов гарантии качества (контрольные перечни) должны храниться в формальной системе управления конфигурацией в соответствии с Планом управления конфигурацией проекта. Процесс гарантии качества, включая процесс аудита, выполняется по указаниям, приведенным в Плане гарантии качества.

Выборка аудита гарантии качества и размер выборки

Аудиты гарантии качества – это аудиты, а не повторные рассмотрения. Как минимум аудиты гарантии качества должны охватывать перечисленные ниже фазы жизненного цикла и артефакты.

Разработка воздушного судна, безопасность полетов, системы, ПО и аппаратура

включая оценки безопасности полетов, требования, проект, реализацию, интеграцию, управление конфигурацией, верификацию и валидацию (Verification and Validation, V&V).

- Планы, если они написаны не службой гарантии качества, должны быть полностью рассмотрены и одобрены ею.
- Все требования, трассируемость, проект, код, тестирование, вспомогательные файлы, записи верификации, двунаправленная трассируемость и процессы, включая критерии перехода, для воздушного судна, систем, ПО и аппаратуры.
- Планы тестирования, процедуры тестирования, тестовые примеры и результаты.
- Вся документация по сторонним инструментам.

- **Планы квалификации инструмента.**
- **Записи управления конфигурацией.**
- **Коллегиальные рассмотрения.**

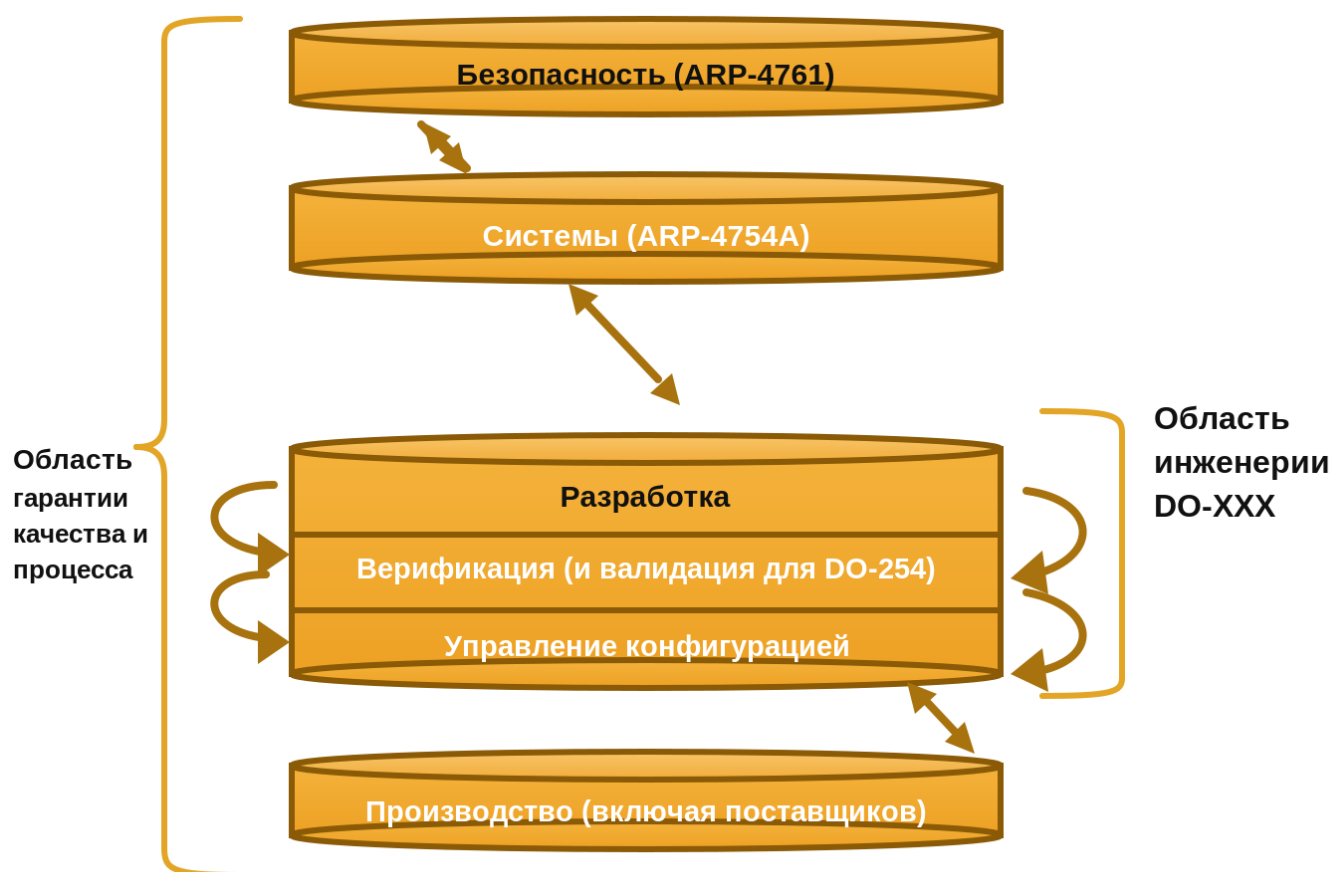
Некоторые компании выделяют гарантии качества достаточно ресурсов и стремятся к 100 %-ному аудиту, но большинству компаний таких ресурсов не хватает, поэтому им приходится проверять только часть артефактов. Хорошее практическое правило – “5-10 %”: гарантия качества проводит аудит примерно 10 % инженерных артефактов. При этом полезно помнить следующее:

- размер выборки аудита должен быть больше 5 %;
- аудит должен охватывать как минимум один артефакт на инженера на фазу; например, если аппаратуру проектируют три инженера, то хотя бы один проект каждого из трех инженеров должен быть проверен аудитом;
- размер и частота выборки аудита должны увеличиваться при росте числа замечаний гарантии качества;
- размер и частота выборки аудита должны быть заранее определены в документации по планированию гарантии качества;
- замечания аудита должны регистрироваться и отслеживаться в процессе обновления; если замечание должно привести к обновлению процесса, помимо обновления артефакта, такая работа также должна отдельно управляться и отслеживаться гарантией качества до завершения.

Для небольших проектов размер выборки аудита должен увеличиваться до 20 % и более.

Получает ли компания дополнительный зачет в разработке авионики, если у нее уже есть сильная корпоративная культура гарантии качества? Ответ однозначный: нет. В авиации каждая версия продукта обычно рассматривается как самостоятельная сущность. Качество каждого продукта оценивается по характеристикам связанных с ним аспектов разработки, а не по более расплывчатым корпоративным признакам, которые могут быть к делу вообще не привязаны. Например, если компания имеет почти безупречные возможности гарантии качества мирового уровня, это еще не значит, что они были полностью применены к конкретному продукту. Всегда возможно, что где-то срезали углы, и компания с сильной корпоративной гарантией качества не использовала эти возможности для данного продукта в полном объеме. Поэтому экосистема разработки авионики живет с философией “виновен, пока не доказана невиновность”, и эта философия распространяется на гарантию качества. При этом компаниям с сильной корпоративной культурой гарантии качества авиационные руководства обычно кажутся достаточно прямолинейными и удобными для внедрения. Особенно если такая культура охватывает обучение инженеров, управление работами, идентификацию и хранение артефактов, формальные рассмотрения и надлежащие средства управления конфигурацией с ведением записей. Опыт показывает, что компании с сильной культурой гарантии качества уже применяют 70 % и более необходимых критериев гарантии качества из документов серии DO.

В типовых авиационных проектах отдельные инженеры сосредоточены на разработке в своей области специализации: безопасность полетов, системные требования, архитектура, ПО, аппаратура или верификация. Руководитель проекта и технический директор держат в поле зрения общую картину и направляют работу отдельных инженерных групп. Но поскольку гарантия качества независима от инженерной организации, сама эта служба должна полностью понимать еще более широкую картину. Гарантия качества должна обеспечить, чтобы все элементы безопасности полетов, системной инженерии, разработки, верификации, управления конфигурацией и производства были определены, оценены и зафиксированы в записях. В наглядной форме эти взаимосвязи лучше всего показаны на следующей сокращенной схеме.



С ростом технической сложности в авиации возрастает и значение гарантии качества: она должна обеспечивать надлежащее управление этой сложностью. Документы серии DO содержат хорошие ориентиры для инженерных процессов, но в части соответствующих ориентиров для гарантии качества они заметно сдержаннее. Почему критерии гарантии качества сформулированы так расплывчато? Напомним: роль гарантии качества состоит в том, чтобы обеспечить достаточность инженерных планов и стандартов, которые определяют реализацию документов серии DO, а затем оценить соответствие инженерной организации этим планам и стандартам. Документы серии DO – это не книга рецептов, предписывающая, как именно определять или выполнять процессы. За пределами определения и оценки инженерных входных и выходных данных мероприятия гарантии качества зависят от множества переменных, сильно различающихся от проекта к проекту: размера, охвата, сложности, инструментов,

критичности, автоматизации и т. д. Задача гарантии качества – учесть эти переменные и определить соответствующие измеримые критерии процесса в инженерных планах и стандартах. Многие современные направления все чаще затрагивают гарантию качества, включая всеобщее управление качеством, постоянное совершенствование, командные улучшения, гибкие методы и т. д. Все они могут быть полезны и должны учитываться в нормальной эволюции гарантии качества. Однако, поскольку они в значительной степени субъективны, базовая сущность гарантии качества в авиации остается неизменной: обеспечить, чтобы инженерные процессы были достаточно определены, а затем соблюдались, давая в результате доказуемое качество систем и воздушных судов.

Заключительное напоминание о гарантии процесса для авионических систем и аппаратуры

ARP4754A задает структуру разработки воздушного судна и систем. DO-254 в основном посвящен инженерным процессам создания сложной электронной аппаратуры – аппаратной логики, реализованной в интегральных схемах, которую раньше часто называли термином *firmware*. Процессы для воздушного судна, систем и аппаратуры похожи на процессы для ПО в DO-178 по простой причине: DO-178 использовался как справочный документ при подготовке руководств для других предметных областей. Однако аспекты гарантии качества для воздушного судна, систем и аппаратуры имеют три принципиальных отличия от ПО:

1. Разработка воздушного судна, систем и аппаратуры охватывает более широкий диапазон процессов жизненного цикла, поэтому в ARP4754A и DO-254 “гарантия качества” называется “гарантией процесса”.
2. Поставщики в разработке воздушного судна, систем и аппаратуры сильнее влияют на качество продукта, поэтому ARP4754A и DO-254 требуют более тщательного аудита поставщиков.
3. Повторяемое производство как часть разработки аппаратуры вносит больше неопределенности, чем в случае ПО, поэтому ARP4754A и DO-254 требуют более тщательного аудита производственных процессов.

По этим причинам гарантия качества для аппаратуры, а также для воздушного судна и систем по ARP4754A, называется гарантией процесса: нужно определить и затем проверять аудитами более широкий диапазон процессов. Дополнительно должны выполняться аудиты, подтверждающие, что объект воздушного судна, системы или аппаратуры, используемый для оценки соответствия, изготовлен в соответствии с соответствующими данными жизненного цикла воздушного судна, системы или аппаратуры. Поэтому должна выполняться инспекция, подтверждающая, что завершенное воздушное судно, система или аппаратура соответствует своим проектным данным. Пример такого мероприятия – инспекция первого изделия.

Соответствие требованиям в военной авиации

Автор: Вэнс Хилдерман

Рецензенты:

- Эмметт Дигнан, старший инженер по авионике, Northrop Grumman
- Винченцо Мирра, менеджер по качеству и сертификации аэрокосмической продукции, Leonardo Company

Оборонные организации во всем мире все активнее применяют DO-178C для ПО, DO-254 для аппаратуры и ARP4754A/ARP4761 для безопасности воздушного судна и систем. Почему это происходит и к чему это приводит?

На протяжении десятилетий военные организации разрабатывали аппаратуру и ПО по специализированным оборонным стандартам, включая 2167A, 498 и 882. Для военных было естественно пользоваться стандартами на аппаратуру и ПО, отличавшимися от коммерческого сектора: считалось, что военные применения “другие”. Главным приоритетом была миссия. Однако сегодня все заметнее сближение военной и коммерческой авионики: *во всем мире внедряются DO-178 и DO-254 вместе с руководящими материалами по системам и безопасности ARP4754A и новой редакцией ARP4761A*. Сегодня соответствие DO-178, а все чаще также DO-254 и ARP4754A/ARP4761, требуется для истребителей (Joint Strike Fighter, T-50 и т. д.), транспортных самолетов и самолетов-заправщиков (C-130, C-17, A400M, KC46 и т. д.), а также для беспилотных летательных аппаратов и беспилотных авиационных систем (Unmanned Aerial Vehicles / Unmanned Aircraft Systems, UAV/UAS), ранее официально называвшихся дистанционно пилотируемыми авиационными системами (Remotely Piloted Aircraft Systems, RPAS).

Что такое DO-178 и DO-254?

Как уже говорилось в предыдущих главах, DO-178C – четвертая редакция стандарта Федерального управления гражданской авиации США (Federal Aviation Administration, FAA) для ПО авионики. Он обязателен для всего коммерческого бортового ПО, влияющего на безопасность полета, поскольку с достаточной степенью уверенности подтверждает: ПО выполняет назначенные ему функции, заданные системными требованиями. Более двадцати пяти лет коммерческое авиационное ПО проходило сертификацию сначала по DO-178, затем по DO-178A, затем по DO-178B, а теперь по DO-178C. Но в начале 2000-х сертифицирующие органы поняли, что безопасность авионики определяется не только ПО, но и аппаратурой. Аппаратура была столь же важна,

как ПО, но от нее требовалось лишь соответствие DO-160, стандарту испытаний на внешние воздействия. Поэтому был создан специальный комитет 180 (Special Committee 180, SC-180), предшественник DO-254, который ввел согласованные сертификационные требования к аппаратуре. Основой DO-254 стал DO-178B: это обеспечило сходство сертификации ПО и аппаратуры по процессам и целям, которые должны быть достигнуты. На более высоком уровне ARP4754A применяется к гражданским воздушным судам и системам, а ARP4761, вместе с новой редакцией ARP4761A, содержит соответствующие подробности анализа безопасности.

DO-178 (ПО) и DO-254 (аппаратура) исходят из того, что ПО и аппаратура должны работать согласованно, причем надежность каждой стороны должна быть доказана. Раньше аппаратура считалась “видимой” и испытывалась на системном уровне вместе с интегрированным ПО; поэтому на нее не распространяли атрибуты качества DO-178. Но это исключение привело к тому, что функциональность переносили из ПО в аппаратуру, чтобы избежать сертификации аппаратуры. Кроме того, сложность аппаратуры выросла настолько, что из-за встроенной логики в программируемых логических устройствах (Programmable Logic Device, PLD), специализированных интегральных схемах (Application-Specific Integrated Circuit, ASIC) и ПЛИС (Field-Programmable Gate Array, FPGA), она часто не уступает ПО, а иногда и превосходит его по сложности. Теперь очевидно, что аппаратура и ПО образуют неразрывную цепь, качество которой определяется самым слабым звеном. Отсюда и требование применять DO-254 также к аппаратуре авионики.

DO-178, DO-254, ARP4754A и ARP4761 используют пять уровней критичности: от уровня А, самого критичного, до уровня Е, наименее критичного. Официально они называются уровнями гарантии разработки (Development Assurance Levels, DAL). Каждой системе авионики назначается один или несколько уровней критичности на основе оценки безопасности системы, где анализируется возможный вклад этой системы в безопасность воздушного судна. Каждый компонент аппаратуры и ПО в такой системе должен соответствовать назначенному уровню критичности или превосходить его. Чем выше уровень критичности, тем строже требования к документации, проектированию, рассмотрению, реализации и верификации.

Военные и гражданские воздушные суда

Раньше военные организации во всем мире использовали собственные стандарты разработки аппаратуры и ПО. Обоснование было таким:

- *военные проекты сложнее коммерческих проектов;*
- *выполнение миссии всегда крайне желательно и в некоторых случаях важнее безопасности;*
- *военным проектам нужно более высокое качество, чем гражданским;*
- *военным проектам приходится управлять многочисленными поставщиками;*

- *военным проектам нужны специализированные военные или чувствительные функции и сложные циклы интеграции;*
- *военные проекты должны учитывать длительный срок службы планера.*

Следует признать, что до появления DO-178, в 1970-х и 1980-х годах, эти доводы были справедливы. Однако к 1990-м они постепенно утратили силу. Сегодня у военной и коммерческой авионики много общего:

- *и там и там высокая сложность и сложная интеграция;*
- *и там и там сотни поставщиков, причем многие поставляют почти одинаковую авионику и военным, и коммерческим заказчикам, а проекты живут долго;*
- *и там и там нужен доступ к передовым коммерческим технологиям;*
- *и там и там все больше внимания уделяется повторному использованию, качеству и экономической эффективности;*
- *и там и там требуется высокий уровень работоспособности, надежности, ремонтпригодности и безопасности;*
- *военные воздушные суда все чаще эксплуатируются в коммерческом воздушном пространстве, где нежелательны ограничения по маршрутам и времени полетов.*

К началу 2000-х военные организации США поняли, что у коммерческого аэрокосмического сектора, особенно в части, регулируемой FAA через DO-178, есть преимущества, которых нет в оборонной среде. Перед ними был выбор:

1. Сохранить статус-кво и ничего не делать.
2. Обновить собственные военные стандарты, включив в них лучшие элементы DO-178.
3. Просто принять DO-178B, а позднее DO-254 и DO-178C.

Какой вариант выбрали? Вариант 3. Было ли это простым решением? Нет. Как и в любой устоявшейся организации, были разные мнения, укоренившиеся практики, сопротивление, дополнительные начальные затраты на переход и, конечно, политика – куда же без нее. Итогом стало постепенное внедрение DO-178, DO-254, ARP4754A и ARP4761A, хотя принятие DO-254, ARP4754A и ARP4761 в оборонном секторе отстает от DO-178. Дополнительно принятие гражданских авиационных руководящих материалов усложняли следующие факторы:

- *военные не хотели передавать надзор FAA, а у FAA не было ни ресурсов, ни полномочий вмешиваться в военные проекты;*
- *военные организации плохо знали специфику DO-178 и DO-254 и поэтому применяли очень разные, субъективные критерии. Если говорить прямо, DO-178 и DO-254 лаконичны и местами туманны; для их практического применения обычно требуется специализированное обучение у экспертов;*

- по указанным причинам военные часто еще сильнее усложняют внедрение DO-178/DO-254, требуя одновременно соблюдать собственные военные стандарты. Намерение понятное, но результат контрпродуктивный: стандарты различаются и в ключевых областях конфликтуют. В DO-178 и DO-254 и без того достаточно неоднозначности и субъективности, а параллельное требование соблюдать военные стандарты резко усложняет задачу;
- военная безопасность обычно строилась вокруг военного стандарта США MIL-STD-882, который несколько отличается от ARP4754A/ARP4761A.

Примеры военных воздушных судов с соответствием документам серии DO

Сегодня многие военные воздушные суда уже достигли как минимум частичного соответствия DO-178 и DO-254. Насколько известно автору, ни одно из них не достигло значимого соответствия ARP4754A и ARP4761A. Не потому, что разработчики планера пытались уйти от ARP47XX, а потому, что во время разработки соответствующих планеров ARP47XX просто еще не был “на радаре”. Вместо этого несколько таких воздушных судов соответствовали военному стандарту США MIL-STD-882E, о чем ниже сказано подробнее. Ниже приведены примеры военных воздушных судов с как минимум частичным соответствием DO-178B/C, а в некоторых случаях и DO-254:

Примеры военных воздушных судов с соответствием документам серии DO

- Транспортные самолеты C-130, C-17, B1, B2.
- Истребитель F-35.
- БПЛА Global Hawk и Predator.
- Самолет-заправщик KC-146.

В отличие от военных стандартов, DO-178 и DO-254 используют пять уровней критичности. Почему? Из-за стоимости. Только из-за стоимости. Если бы стоимость не имела значения, все ПО авионики относили бы к уровню А, самому критичному уровню с самыми строгими требованиями. Но десятки систем авионики на борту воздушного судна влияют на безопасность полетов не одинаково. Уровень критичности выбирается с помощью аналитических процессов, которые оценивают вклад каждой системы, подсистемы и компонента в безопасность воздушного судна. Этот уровень также опирается на инженерное суждение, летный опыт и срок службы системы. Такие анализы безопасности описаны в специальных стандартах, включая ARP4761, а с 2020 года – ARP4761A для DO-178/DO-254, и хорошо известны в гражданской авиации. Для военной авионики они относительно новы, отсюда и субъективность при выборе уровня критичности в оборонных проектах.

Что касается стоимости, следующий график показывает прирост стоимости для разных уровней критичности:

График: прирост стоимости и сроков DO-178 по уровням критичности

Уровень критичности – стоимость



Стоимость и сроки по уровню критичности напрямую зависят от количества и сложности целей, которые нужно выполнить. Но важны и другие факторы: что именно заявитель, будь то отрасль или организация, понимает под применимыми руководящими материалами серии DO и как он их интерпретирует. Именно здесь часто возникают недопонимание, ошибки, перепланирование и переделки. Это одна из главных причин роста затрат на авиационное соответствие.

Распространенный миф: DO-178 дорог сам по себе. Как видно выше, сертифицированное ПО уровня D все равно требует процессов планирования, работы с требованиями, реализации, рассмотрений и базового тестирования. Кроме того, для уровня D применяются управление конфигурацией, гарантия качества и взаимодействие с представителями FAA по инженерной экспертизе (Designated Engineering Representatives, DER). Но затраты уровня D едва ли выше затрат любого несертифицируемого коммерческого процесса разработки ПО. Почему? Потому что уровень D почти полностью состоит из обычных отраслевых принципов программной инженерии.

Еще один миф: основной скачок затрат происходит при переходе с уровня B по DO-178 на уровень A. Это неверно.

Наибольшее влияние DO-178 на стоимость проявляется при переходе с уровня D на уровень C. Почему? Уровень C требует следующего, чего нет на уровне D и что увеличивает бюджет и сроки как минимум на 30 % по сравнению с уровнем D:

- тестирование требований к ПО низкого уровня;
- обеспечение 100 % покрытия всех операторов;
- более строгие рассмотрения;
- во многих случаях более строгое управление конфигурацией;
- гораздо более полная трассируемость, чем в традиционных военных стандартах; в большинстве проектов для этого используют инструмент.

Уровень В требует дополнительного структурного покрытия, то есть покрытия решений/условий – всех ветвей в исходном коде, дополнительной независимости рассмотрений и более жесткого управления конфигурацией. На первый взгляд кажется, что уровень В должен быть существенно дороже уровня С, примерно на 50-70 %. В теории это выглядит логично. Но, как нередко бывает, практика снова портит красивую теорию. На уровнях В и С должны существовать подробные требования к ПО низкого уровня, и они должны быть тщательно протестированы. При таком тестировании на основе требований подавляющее большинство ветвей, обычно 70-90 %, уже выполняется и не требует дополнительного тестирования структурного покрытия, если корректно применяются средства захвата тестов и анализа покрытия. Поэтому кажущийся большой прирост стоимости структурного покрытия для уровня В по сравнению с уровнем С уже частично снимается тестированием на основе требований. Кроме того, зрелые организации по разработке ПО уже имеют полуавтоматизированный, упорядоченный процесс с независимыми рассмотрениями и жестким управлением конфигурацией. Следовательно, дополнительные затраты на эти аспекты уровня В в значительной степени сглаживаются. Читателю настоятельно рекомендуется заранее пройти обучение по DO-178 и выполнить совершенствование процесса DO-178, чтобы использовать эти методы снижения затрат.

Уровень А – самый критичный уровень ПО и, следовательно, самый дорогой. Это верно. Но есть еще один миф об уровне А: *“Достичь уровня А чрезвычайно трудно, и он будет стоить как минимум на 30-50 % дороже уровня В”*. Это неверно. Уровень А добавляет требования к структурному покрытию, включая модифицированное покрытие условий/решений (Modified Condition/Decision Coverage, MC/DC), требования к робастности и корреляции, а также немного более строгие рассмотрения. Главный фактор роста стоимости по сравнению с уровнем В – именно требование MC/DC. Однако при правильном применении современных инструментов структурного покрытия, обучении персонала и тщательном тестировании на основе требований дополнительные затраты на уровень А можно в значительной степени сдержать.

Указанные различия в стоимости – реальные и достижимые результаты, документально подтвержденные автором на десятках успешных проектов, а также у ведущих аэрокосмических поставщиков. Но эти значения НЕ являются средними по отрасли: средний авиационный проект по DO-178 превышает эти приросты стоимости на 20-50 %. Почему?

Из-за неэффективности, непонимания DO-178 и неприменения лучших практик сдерживания затрат. Все это приводит к переделкам и лишней работе.

Безопасность: ARP4761 и ARP4754A

Традиционно военные применяли собственные стандарты безопасности, например широко распространенный военный стандарт США MIL-STD-882E. Однако военные стандарты безопасности плохо сочетаются с гражданскими руководящими материалами по ПО и аппаратуре. Масло с водой смешать тоже можно, но обычно недолго и без особой пользы. ARP4761 и ARP4754A гораздо лучше интегрируются с DO-178C, поскольку опираются на назначение уровня гарантии разработки (Development Assurance Level, DAL), особенно в подходах с уровнем гарантии разработки функции (Function Development Assurance Level, FDAL) и уровнем гарантии разработки элемента (Item Development Assurance Level, IDAL). Кроме того, гражданское применение предварительной оценки безопасности воздушного судна (Preliminary Aircraft Safety Assessment, PASA) и предварительной оценки безопасности системы (Preliminary System Safety Assessment, PSSA) достаточно специфично и в MIL-STD-882E освещено не так полно. См. отдельные материалы AFuzion по ARP4754A и ARP4761. С выходом обновленной ARP4761, то есть ARP4761A, в 2021 году ожидалось, что военные организации по всему миру будут полнее переходить на гражданские аналоги.

Соответствие и сертификация: особенности военного соответствия

Поскольку FAA, за очень редкими исключениями, не участвует в военных проектах, формальная сертификация не требуется. Вместо этого военные ведомства обычно сами подтверждают выполнение требований и используют термин *соответствие*. Поэтому военным требуется соответствие DO-178/DO-254, а не сертификация. В чем разница? При таком соответствии:

- *FAA не участвует;*
- *DER не требуются, хотя их участие рекомендуется;*
- *некоторые требования смягчаются, включая анализ безопасности, робастность кода, строгое доказательство MC/DC во всех случаях и т. д.;*
- *военные чаще отказываются от отдельных целей DO-178C и DO-254 либо снижают порог их приемки, включая критерии квалификации инструмента, тестирование на робастность, детализацию требований низкого уровня и соблюдение критериев перехода.*

Краткий обзор других стандартов: MIL-STD-498, MIL-STD-882E и ISO/IEEE 12207

Сегодня военные во всем мире постепенно переходят к ARP4761/A и ARP4754A, как кратко описано выше и в предыдущих специализированных главах. Однако большинство военных воздушных судов раньше применяли стандарты не из серии ARP47XX: военный стандарт США MIL-STD-498, военный стандарт США MIL-STD-882E и ISO/IEC/IEEE 12207, стандарт процессов жизненного цикла ПО (Systems and software engineering – Software life cycle processes; в исходнике – ISO/IEEE 12207). Откровенно говоря, по мнению автора, эти стандарты хороши и в целом похожи на гражданские стандарты, особенно на ARP4761 в части безопасности. Но, по опыту автора, стандарты вне ARP47XX и серии DO более предписывающие, а значит, более субъективные: их труднее единообразно применять и оценивать. Ниже приведен очень краткий обзор других основных стандартов, которые раньше применялись на унаследованных военных воздушных судах.

Кратко о MIL-STD-498

Ниже кратко изложено содержание MIL-STD-498:

- военный стандарт для систем вооружения, включая наземные системы, в отличие от DO-178C, которая относится только к бортовым системам;
- 22 описания единиц данных (Data Item Descriptions, DID), которые нужно подготовить для каждого проекта; эти описания предписывают форматы документов, то есть задают “как делать”;
- ПО состоит из программных единиц (Units), группы единиц называются компьютерными программными компонентами (Computer Software Components, CSC), а группы таких компонентов – единицами конфигурации компьютерного ПО (Computer Software Configuration Item, CSCI), обычно это один исполняемый файл;
- хотя каскадная модель не обязательна, типичные процессы MIL-STD-498 выглядят как каскадная модель;
- в отличие от DO-178C, MIL-STD-498 также предписывает управление, оценку рисков и учет внешней экосистемы, чего нет в DO-178C;
- как и DO-178C, требует стандартов для требований, проекта, кода и тестов, хотя в DO-178C нет отдельных стандартов тестирования;
- в отличие от DO-178C, уделяет внимание аппаратным ресурсам, например памяти, и требованиям к ПО более высокого приоритета;
- в отличие от DO-178C, содержит подробности по инкрементным сборкам и практическому выполнению больших сложных программ, включая управление и проектные риски;

- делает больший упор на тестирование “черного ящика”; в нем отсутствуют характерные для DO-178C анализ потоков данных, анализ потоков управления, структурное покрытие, строгие критерии перехода, тестирование на робастность, оценка безопасности, квалификация инструмента и взаимодействие с сертифицирующим органом.

Кратко о MIL-STD-882E

Ниже кратко изложено содержание MIL-STD-882E:

- силен в анализе безопасности, как ARP4761;
- силен в анализе рисков, как план программы безопасности ARP4754A;
- использует субъективные критерии оценки рисков;
- силен в требованиях безопасности, валидации, верификации и отслеживании;
- силен в управлении конфигурацией;
- слаб в процессах ПО и стандартах проектирования/кодирования;
- силен в тестировании ПО и системы.

Кратко об ISO/IEEE 12207

Ниже кратко изложено содержание ISO/IEEE 12207:

- в значительной степени “заимствован” из DO-178C;
- как и DO-178C, использует программные процессы, а не последовательные этапы;
- дает сильную модель систем и ПО, подходящую для критически важных с точки зрения безопасности систем;
- унаследовал многое от модели зрелости возможностей (Capability Maturity Model, CMM), а затем от интеграции моделей зрелости возможностей (Capability Maturity Model Integration, CMMI): силен в доказательствах и совершенствовании;
- силен во внутренних аспектах ПО, а также в верификации “черного ящика”;
- силен в проекте ПО и коде, включая временные характеристики и робастность;
- слаб в объективности и гораздо более субъективен, чем DO-178C;
- силен в управлении конфигурацией, свидетельствах, верификации и валидации (Verification and Validation, V&V);

- слаб в квалификации инструмента, анализе потоков данных, потоков управления и межкомпонентной связности, структурном покрытии по требованиям, взаимодействии с сертифицирующим органом; не имеет требований высокого и низкого уровня и использует менее формальные переходы;
- гораздо сильнее DO-178C в части внешних процессов, таких как управление, метрики, бизнес и т. д.

Анализ разрывов для военных организаций, переходящих на документы серии DO и ARP47XX

У большинства военных организаций и поставщиков уже есть в целом качественные организации и процессы. При внедрении документов серии DO и ARP47XX они могут повторно использовать значительную часть существующих процессов, документации и артефактов. Нередко они фактически уже находятся на уровне 60-70 % соответствия DO-178 и 30-50 % соответствия DO-254, даже если специально не рассматривали DO-178/DO-254. Поэтому при требовании “соответствовать документам серии DO” наиболее экономично выполнить анализ разрывов и закрыть выявленные пробелы, а не начинать все заново. Такой анализ оценивает текущие процессы и определяет разрывы относительно полного внедрения DO-178 или DO-254. Обычно анализ разрывов занимает 2-4 человеко-недели, если его выполняют опытные эксперты по DO-178/DO-254, и может сэкономить годы за счет максимального повторного использования. В отличие от военных стандартов, которые жестко задают формат и содержание документов и артефактов, DO-178/DO-254 дают больше свободы, а значит, позволяют сохранять и повторно использовать существующие элементы.

Когда автор выполняет анализ разрывов DO-178/DO-254 для военного заказчика, в проверяемых организациях обычно обнаруживаются следующие уровни разрывов, где “0 % разрыва = 100 % соответствия”:

- **Организация уровня 1 по модели зрелости возможностей Института программной инженерии (Software Engineering Institute Capability Maturity Model, SEI CMM): разрыв 70-90 %**
- **Организация уровня 2 по SEI CMM: разрыв 50-75 %**
- **Организация уровня 3 по SEI CMM: разрыв 35-60 %**
- **Организация уровня 4 по SEI CMM: разрыв 25-40 %**
- **Организация уровня 5 по SEI CMM: разрыв 20-35 %**

Из этого следуют два неожиданных факта. Во-первых, даже у организаций уровня 5 по CMMI разрывы все еще значительны, потому что CMMI не включает такие базовые для DO-178C элементы, как два уровня требований к ПО, квалификация инструмента, структурное покрытие (покрытие операторов, покрытие решений, MC/DC), экстремальное тестирование на робастность, анализ потоков данных, потоков управления и

межкомпонентной связности и т. д. Во-вторых, разрывы для уровня 3 по СММІ сильно различаются: разрыв 60 % у уровня 3 появляется тогда, когда реальная инженерная практика больше похожа на уровни 1-2, чем на уровень 3.

На уровне отдельных мероприятий DO-178/DO-254 типичные разрывы выглядят так:

МЕРОПРИЯТИЕ ПО СЕРТИФИКАЦИИ	% РАЗРЫВА
<i>План сертификации ПО (Plan for Software Aspects of Certification, PSAC)</i>	80 %
<i>План гарантии качества</i>	20-30 %
<i>План управления конфигурацией</i>	10-20 %
<i>План разработки</i>	40-50 %
<i>План верификации ПО</i>	60-70 %
<i>Оценка безопасности</i>	80-90 %
<i>Определение требований</i>	20-30 %
<i>Проектирование</i>	10-15 %
<i>Код</i>	5-10 %
<i>Функциональное тестирование</i>	5-10 %
<i>Тесты структурного покрытия</i>	90-100 %
<i>Управление конфигурацией</i>	10-30 %
<i>Гарантия качества</i>	50 %
<i>Квалификация инструмента</i>	100 %
<i>Контрольные перечни</i>	30-50 %
<i>Рассмотрения</i>	30-50 %
<i>Аудиты</i>	30-50 %
<i>Взаимодействие с DER</i>	100 %

Типичные разрывы в новых военных проектах DO-178C и DO-254 уровня гарантии разработки В по сравнению с уровнем 3 СММІ

Преимущества DO-178/DO-254/ARP4754A в военных проектах

DO-178 не бесплатен, как отмечалось выше. Но при правильном понимании и внедрении DO-178 может быть экономически эффективным даже в военных проектах. Почему же тогда так много военных организаций внедряют DO-178/DO-254? Потому что преимущества действительно есть. Ниже перечислены наиболее часто получаемые преимущества DO-178/DO-254/ARP4754A для военных проектов, основанные на опыте автора, успешно работавшего более чем на 150 аэрокосмических проектах.

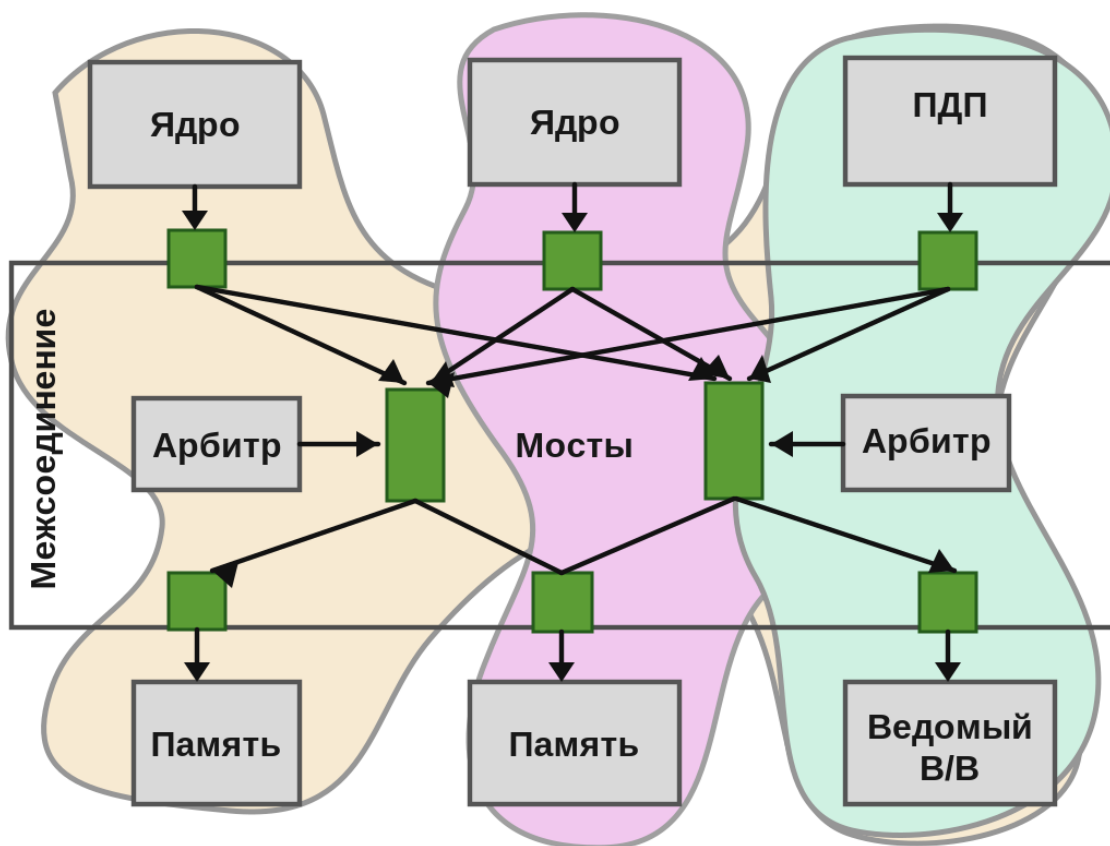
- *Более высокая прозрачность поставщиков.* В DO-178 и DO-254 расширенные процессы подготовки артефактов и рассмотрений дают заказчику лучшую видимость работы поставщиков.

- *Более ясные требования на раннем этапе.* ARP4754A требует требований безопасности и системных требований, охватывающих всю функциональность на уровне системы, безопасность, характеристики, производные требования и интерфейсы с аппаратурой и ПО. DO-178 требует полноценных и подробных требований к ПО, как высокого, так и низкого уровня. Такая детализация и дисциплина заставляют отвечать на вопросы заранее, а не откладывать их. Допущения резко сокращаются. Обеспечиваются согласованность требований и возможность их тестирования. Значительно уменьшается число итераций и переделок из-за ошибочных или отсутствующих требований.
- *Меньше итераций реализации.* Итерационные переделки реализации и кода – беда инженерии ПО и аппаратуры. Во многих новых продуктах существует 10, 20 и даже 30 версий развивающихся файлов кода. Это бессмыслица. Код и логика должны быть в основном правильными уже при первом написании и не должны требовать десятков обновлений, чтобы стать правильными. Код следует рассматривать через анализ реализации относительно документированных требований.
- *Меньше критических единичных отказов проекта.* ПО – это искусство; художники сопротивляются документированию своей работы, общим стандартам разработки и экспертным рассмотрениям. Без стандартов, дисциплины и современных принципов программной инженерии команды ПО распадаются на слабо организованную группу независимых художников. Эти люди могут быть очень ценными, творческими и талантливыми. Но потеря или слабость любого такого художника по любой причине катастрофична для команды, если его работа не документирована, не понята и не применяется последовательно так же, как работа остальных участников. DO-178C существенно снижает риск таких критических единичных отказов проекта.
- *Лучшая осведомленность руководства о реальном состоянии графика.* Сколько проектов ПО неделями сообщают статус “готово на 99 %”? Как измерять прогресс по ПО? Как руководство может действительно понять степень готовности ПО? Ответ на все эти вопросы дают современные и достаточно детальные методы управления, построенные вокруг DO-178. DO-178 обеспечивает видимость, трассируемость и точный статус проектирования, разработки, тестирования, интеграции и рассмотрений.
- *Большая согласованность внутри ПО.* ПО похоже на цепь: оно настолько прочно, насколько прочно его самое слабое звено. ПО, которое правильно на 99 %, неправильно на 1 %, а значит, небезопасно. Самый слабый программный модуль или инженер по ПО находится на критическом пути безопасности ПО. Все ПО должно быть согласованным со своим уровнем критичности, и DO-178 заставляет это выполнять.

- *Меньше дефектов, обнаруживаемых при интеграции.* Интеграция может быть долгим итерационным процессом, где выявляются и исправляются крупные дефекты, требующие изменений в проекте. Но не при DO-178: там интеграция обычно проходит на 50-75 % быстрее, чем в средах без DO-178.
- *Лучшее повторное использование.* Подробная и согласованная документация, требуемая DO-178C, модульность, соблюдение документированных современных инженерных принципов и рассмотрения, подтверждающие все перечисленное, существенно улучшают повторное использование. В ПО повторное использование – святой Грааль. Но реальность такова: если программный компонент нельзя повторно использовать хотя бы на 80 %, то есть без изменений, быстрее и менее рискованно начать заново. Большая часть ПО повторно используется менее чем на 50 %. При DO-178C, соблюдении стандартов проектирования и кодирования, независимых рассмотрениях и трассируемости большинство модулей должно быть повторно используемо как минимум на 90 %.
- *Более простое регрессионное тестирование.* Продукт создают в расчете на успех и долгий срок службы. И ясно, что ПО будет развиваться через новые применения, установки и версии. Регрессионное тестирование может быть дорогим, если оно обширное или выполняется вручную. DO-178 дает полноценную трассируемость, чтобы определить, какие модули требуют изменений или анализа и соответствующего повторного тестирования.
- *Лучшая интеграция с аппаратурой.* Интеграция ПО на целевую аппаратную систему обычно сложна для встроенных систем: среда разработки сильно отличается от целевой среды, а реализуют их разные инженерные команды. DO-178 требует тестировать на аппаратуре те компоненты ПО, на которые аппаратура может влиять: интерфейсы аппаратуры и ПО, прерывания, временные характеристики, компоненты уровня платы, пакеты поддержки платы (Board Support Package, BSP), операционные системы реального времени, ОСРВ (Real-Time Operating System, RTOS) и т. д. Повышенные детерминизм и качество всех этих компонентов благодаря DO-178 улучшают аппаратную интеграцию.
- *Более сильный фокус на системе и безопасности.* Военные проекты все чаще следуют гражданскому требованию учитывать ARP4754A для системных аспектов и ARP4761/A для безопасности. Если раньше военные часто применяли военный стандарт США MIL-STD-882E (2012), документ Министерства обороны США “Стандартная практика: безопасность системы” (Standard Practice: System Safety), то сегодня они все чаще согласуют свои подходы с гражданскими аналогами по системам и безопасности – ARP4754A и ARP4761/A. То же уже произошло с ПО и аппаратурой через DO-178C и DO-254 соответственно. Когда требования к системам и безопасности применяются заранее на уровне воздушного судна, затем на уровне систем, а далее итерационно и параллельно на протяжении всего проекта, улучшаются и разработка авиационной техники в целом, и соблюдение требований безопасности.

Многоядерные процессоры для авионики по CAST-32A (с интегрированной модульной авионикой (ИМА) по DO-297)

Авторы: Марк Браун, ведущий архитектор, Lynx Software, и Вэнс Хилдерман, технический директор AFuzion Inc.



Многоядерная обработка (MCP) и CAST-32A

У новичков в авиационной разработке и сертификации часто уже есть мнение, что эта область статична и несколько застывшая. И, честно говоря, если смотреть только на частоту обновления отдельных руководств серий DO-XXX и ARP47XX, которые меняются раз в 10-20 лет, такое мнение кажется справедливым. А COVID-19 – это просто грипп... конечно же нет.

Вычислительная техника, возможно за исключением недавних успехов биотехнологий, является одной из самых быстро развивающихся областей в истории нашей планеты. Неудивительно, что авиационные вычислительные технологии тоже быстро развиваются. Да, вычисления в гражданской авиации всегда будут отставать и от самого переднего края,

и даже от просто передовых решений. Так требует безопасность общества. В предыдущих главах этой книги рассматривались разработка на основе моделей, объектно-ориентированная технология и современные инструменты для ПО. Четыре десятилетия назад ничего из этого не существовало, а сегодня это обычная практика.

Люди редко бывают единодушны по какому-либо вопросу, за возможным исключением вычислительной мощности: ее постоянно нужно больше. В прошлом изготовители полупроводников просто сокращали расстояние прохождения электрического сигнала и оптимизировали одноядерные процессоры, чтобы повысить скорость обработки. Но, как и с конечным запасом воды, в какой-то момент возможность наращивать пропускную способность иссякает. Именно это произошло со вчерашними одноядерными микропроцессорами: лучшим способом получить больше вычислительной мощности стало объединение нескольких согласованно работающих процессоров в одном вычислительном устройстве, известном как многоядерный процессор (Multi-Core Processor, MCP).

CAST-32A представляет согласованную позицию органов сертификации авионики в отношении многоядерных процессоров. Хотя современная аэрокосмическая экосистема могла бы выиграть от их применения, на момент публикации CAST-32A Федеральное управление гражданской авиации США (Federal Aviation Administration, FAA) и Агентство Европейского союза по авиационной безопасности (European Union Aviation Safety Agency, EASA) еще не разработали способ получения сертификационного зачета для критически важного с точки зрения безопасности ПО, размещенного на многоядерном процессоре. Для этого позиционный документ CAST-32A определяет темы, требующие рассмотрения и способные повлиять на безопасность, производительность и целостность авиационного ПО по DO-178C, размещаемого на многоядерных процессорах.

Для каждой темы в документе приводится обоснование, объясняющее, почему эта тема вызывает озабоченность, и предлагаются цели для ее устранения. (CAST-32A, раздел “Purpose”, с. 3)

Поскольку соответствующие документы по сертификации авиационного ПО (DO-178B/C и ED-12B/C) были написаны до начала применения многоядерных процессоров на гражданских воздушных судах, эти сертификационные руководства рассматривают только ПО, выполняющееся на одноядерной аппаратуре. Группа сертификационных органов по ПО (Certification Authorities Software Team, CAST) – международная группа авиационных экспертов, которая разъясняет и гармонизирует экосистему авиационной разработки. Позиция CAST-32A состоит в том, что многоядерные процессоры способны дать убедимые преимущества по габаритам, массе, энергопотреблению и стоимости (Size, Weight, Power, and Cost, SWaP-C), а современные поставщики аэрокосмического оборудования заинтересованы в их применении в своих системах.

Мир потребительских устройств полностью принял многоядерные процессоры, и многие устройства, которыми ежедневно пользуются читатели этой статьи, содержат такие процессоры. Некоторые даже предсказывают полное исчезновение одноядерных процессоров (Single-Core Processors, SCP).

Авионика завтрашнего дня почти наверняка будет содержать более сложные системы, а значит, ей потребуется значительно большая вычислительная мощность. Многоядерные процессоры являются одним из основных ответов на эту быстро растущую потребность в усовершенствованных вычислительных архитектурах и вычислительной мощности. Поэтому аэрокосмическая отрасль в целом должна определить, как лучше использовать многоядерные процессоры в будущих проектах. Но как преодолеть связанные с ними сложности?

Прежде чем описывать темы, связанные с многоядерными процессорами и CAST-32A, где основной акцент сделан на обособлении и его деградации из-за интерференции, стоит рассмотреть предпосылки, влияющие на применение многоядерных процессоров. Инженерам и руководителям, которые уже планируют использовать многоядерные процессоры как аппаратные целевые платформы для следующего поколения своего ПО, полезно учитывать эти факторы при проектировании будущих решений.

Одноядерные и многоядерные процессоры

Следующая упрощенная схема показывает основные различия между одноядерными и многоядерными процессорами:

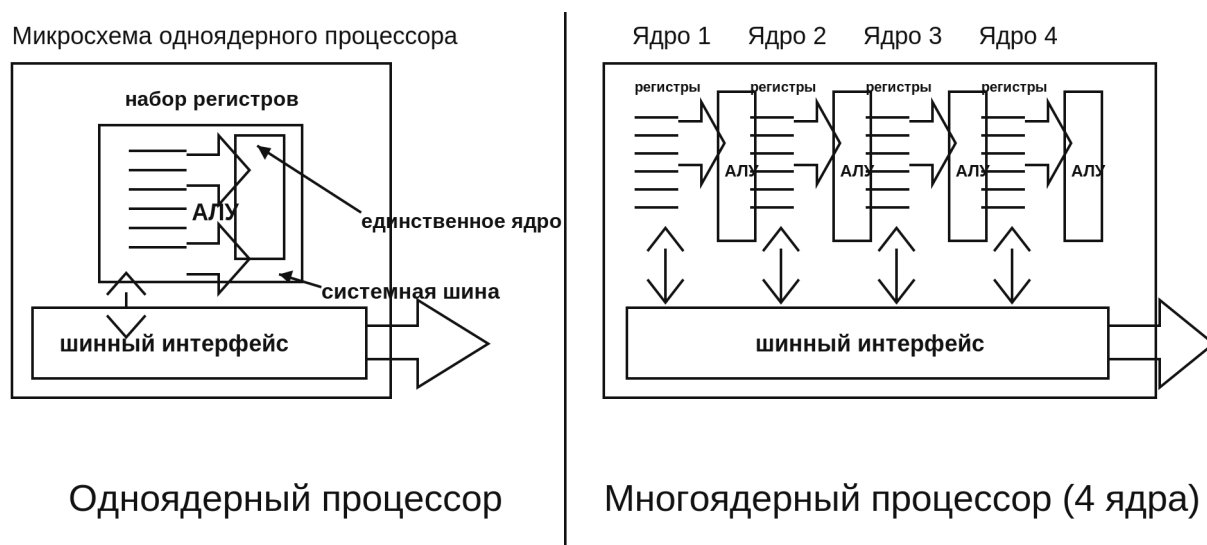


Схема одноядерного и многоядерного процессора (упрощенная)

Как показано выше, каждое отдельное ядро (слева) состоит из набора регистров и связанного с ним арифметико-логического устройства, АЛУ (Arithmetic Logic Unit, ALU). Многопоточные ядра (не показаны) могут проектироваться за счет дублирования наиболее интенсивно используемых частей одного ядра, например путем добавления второго файла регистров. В настоящих многоядерных процессорах (справа) ядра komponуются в

плиточную структуру, образуя большой набор ядер. Обычно и одноядерные, и многоядерные процессоры используют шинный интерфейс для доступа ко все более насыщенной и сложной сети полупроводниковых и/или электронных периферийных ресурсов вне ядра.

Разные разработчики по-разному объясняют применение многоядерных процессоров, но обычно называют следующие преимущества:

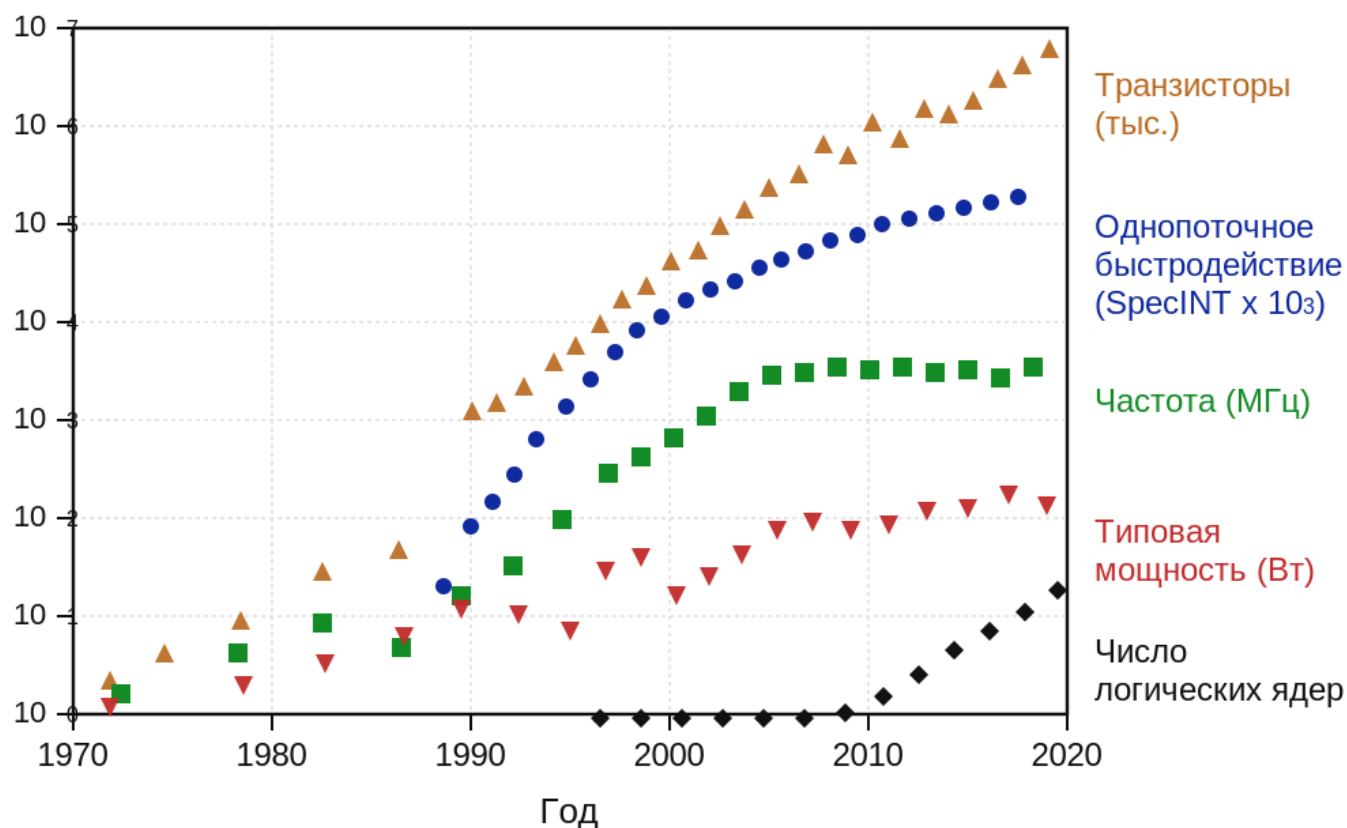
- более высокая производительность;
- одновременное выполнение;
- меньшее энергопотребление и тепловыделение на отдельную выполненную инструкцию;
- использование многопоточных приложений;
- поддержка закона Мура (см. подробности ниже);
- доступ к “новейшим” и часто лучшим в своем классе коммерческим технологиям.

Дом, квартира или гостиница?

Рынки ПК, серверов и мобильных устройств, использующие операционные системы симметричной многопроцессорной обработки (Symmetric Multiprocessing, SMP), с жадностью поглощают массовые компоненты многоядерных процессоров, доступные с 2005 года. Корпоративные серверные просто масштабировались и стали быстрее выполнять более тяжелые нагрузки. В отличие от этих рынков, жесткие сроки реального времени и вопросы безопасности, характерные для ПО, сертифицируемого по DO-178C, задержали внедрение многоядерных процессоров в авиационное ПО. Темы, поднятые в CAST-32A, попадают в самую суть различий между операционными системами SMP, которые легко масштабировались для многоядерных процессоров, и операционными системами реального времени, где этого не произошло.

Эти вопросы подчеркиваются и во многих других публикациях FAA и Радиотехнической комиссии по аэронавтике (Radio Technical Commission for Aeronautics, RTCA). Например, эталонная модель обработки в реальном времени приведена в FAA AR05/27. В этой модели авиационные системы реального времени вычисляют “закон управления”, который с фиксированной периодичностью принимает входы от датчиков и выдает выходы на исполнительные механизмы. Если исполнительный механизм управляет аэродинамической поверхностью или другим критически важным с точки зрения безопасности аспектом воздушного судна, пропуск срока может иметь катастрофические последствия. В таких условиях наиболее сложные проблемы применения многоядерных процессоров связаны с защитой, или обособлением, ПО, отвечающего за соблюдение этих критически важных сроков, а также с обеспечением детерминизма неизменяемой логики производителя, встроенной в многоядерный процессор.

В корпоративном контексте задачи, процессы и потоки обычно чаще перемещают со своего “домашнего” процессора – с темпом, который можно сопоставить с законом Мура. Напротив, перенос встроенных систем реального времени на новую платформу вызывает вопросы безопасности и сертификационные затраты, которые нужно заново учитывать всякий раз, когда авиационное ПО размещается на другой модели процессорного ядра. Как правило, в авиационных и критически важных с точки зрения безопасности контекстах ПО, которому нужны детерминированные сроки реального времени, жестче привязано к конкретному процессорному ядру и целевой плате. Поэтому закон Мура системам реального времени одновременно помогает и мешает. График закона Мура (ниже) помогает рассмотреть эти вопросы через экспоненциальный рост на протяжении десятилетий:



Исходные данные до 2010 г. собрали и построили М. Хоровиц, Ф. Лабонт, О. Шачам, К. Олкотун, Л. Хэммонд и К. Баттен.
Новый график и данные за 2010-2017 гг. подготовил К. Рупп.

42 года трендовых данных по микропроцессорам

Полезную метафору для этого обсуждения можно получить, если представить перенос критически важной с точки зрения безопасности задачи реального времени с ее собственного выделенного однопоточного процессора в “новый дом”. Является ли этот новый дом: (1) постоянно выделенным сооружением, как отдельный дом, (2) сооружением совместного пользования, как квартира, или (3) комнатой с разделением времени, как гостиница?

Значим ли CAST-32A сегодня?

В 2014 году FAA выпустило CAST-32, который позднее был заменен CAST-32A. В то время независимые отраслевые эксперты, например Дэвид Артерберн, директор Центра системной инженерии и моделирования винтокрылых летательных аппаратов Университета Алабамы в Хантсвилле, сдержанно оценивали краткосрочные перспективы многоядерной обработки и указывали на сложность понимания и прогнозирования сложных взаимодействий внутри четырехъядерных микросхем. Тогда Артерберн обобщал 36 исследований различных рабочих групп, финансируемых исполнительным управлением программ армейской авиации США. В. Х. Дова в своей оценке уровней технологической готовности будущей интегрированной модульной авионики второго поколения (Second Generation Integrated Modular Avionics, IMA2G) сообщает:

“Артерберн преуменьшал значение документа CAST-32, утверждая, что ни одна установка многоядерной микросхемы фактически не получила официального подтверждения летной годности. Однако он быстро отметил, что вопрос вскоре станет крайне острым. “В течение пяти лет, – предсказал он, – вы не сможете купить массовый одноядерный процессор” (Д. Р. Артерберн, личное сообщение, 22 января 2015 г.)” (цитируется по Dova, 2015).

График закона Мура (на предыдущей странице) подтверждает прогноз Артерберна. Массовые многоядерные процессоры не только набирают распространение по мере снижения выпуска одноядерных процессоров. Красные треугольники, показывающие число транзисторов, также неизбежно упрутся в предел, обусловленный законами физики (минимальный размер полупроводникового кристалла) и экономической целесообразностью (массовая цена за транзистор). Среди заметных сторонников этого прогноза – Боб Колвелл, бывший главный архитектор IA-32 в Intel и Intel Fellow (по сообщению MIT Technology Review). Гордон Мур с этим согласен (по сообщению Рейчел Кортленд для Института инженеров электротехники и электроники (IEEE, Institute of Electrical and Electronics Engineers)).

Хотя экспоненциальный рост числа транзисторов прекратится, число логических ядер в корпусе микропроцессора, или “чипе”, экспоненциально росло еще до 2010 года. Именно эту новую тенденцию многоядерных процессоров, показанную на графике выше черными ромбами, и рассматривает CAST-32A.

Хотя CAST-32A “предлагает цели для устранения [озабоченности, связанной с многоядерными процессорами]”, плохая новость состоит в том, что выполнение этих целей остается обязанностью тех, кто решит использовать многоядерные микросхемы. Хорошая новость CAST-32A в том, что выполнение этих целей сделает возможным будущее применение многоядерных процессоров в авиационных контекстах реального времени и критически важных с точки зрения безопасности.

Следовательно, значимость CAST-32A проявится, если он позволит применять многоядерные процессоры в критически важных с точки зрения безопасности контекстах. Возвращаясь к нашей метафоре, вопрос состоит в том, как перенести задачи жесткого реального времени из их постоянных выделенных домов на одноядерных процессорах в новые дома на многоядерных процессорах. В то же время CAST-32A можно рассматривать как средство консолидации процессов мягкого реального времени и некритичных процессов на одном и том же многоядерном процессоре, подобно жильцам квартиры или гостиницы. Вероятно, как никогда раньше понадобится архитектура обособления, чтобы будущая консолидация бортовых задач ПО на многоядерных процессорах по-прежнему надежно соблюдала сроки реального времени. Когда многоядерные процессоры используются для консолидации ПО, ранее размещавшегося на отдельных аппаратных процессорах, то есть на нескольких микросхемах, физическое обособление, которое раньше предполагалось и легче доказывалось, теперь должно быть обеспечено явно.

Перенос ПО в новый дом

Инженеры в более широкой области встроенных систем могут подтвердить действие еще одной важнейшей рыночной силы, влияющей на будущие микросхемы. Уже сам выбор между доминирующими на рынке архитектурами команд, например PowerPC, Intel или ARM, открывает ящик Пандоры. Примерно с момента выхода Apple iPhone в 2007 году мировые поставки смартфонов стали значительно превосходить поставки ПК и серверов, к которым десятилетиями применяли закон Мура. Если рассматривать это как широкое влияние на историю полупроводников, iPhone и последующая популярность смартфонов за прошедшее десятилетие вывели эти встроенные системы на место самого массово развернутого типа вычислительных систем в мире. Выбор между массовыми микросхемами PowerPC, Intel или ARM, как правило, затрагивает и периферийные вопросы, связанные с соглашениями по вводу-выводу и подключению сопроцессоров, характерными для суперкомпьютеров, ПК/серверов и мобильных решений.

Поэтому CAST-32A также нужно рассматривать в контексте рыночного влияния системы на кристалле, СнК (System-on-a-Chip, SoC). Популярные микросхемы PowerPC, Intel и ARM используют разные конструкции систем на кристалле и все более широкий набор системных ресурсов. То есть, в отличие от массива одинаковых процессорных ядер, большинство современных массовых микросхем также предоставляет целую гамму разнородных коммуникационных ресурсов – от кэш-памяти и встроенной динамической памяти до устройств ввода-вывода и высокоскоростных межсоединений, которые могут подключать как группы устройств, так и дополнительные вычислительные ядра. Сегодня трудно найти массовую микросхему без системных ресурсов вне ядра: управления питанием, управления памятью, средств отладки и управления тактированием, контроллеров прерываний, встроенных устройств ввода-вывода и т. д.

Вопросы, которые нужно рассматривать с точки зрения безопасности, включают как процессорные ядра многоядерного процессора, так и системные ресурсы вне ядра. С учетом более широких вопросов безопасности, связанных с понятием интерференции, CAST-32A рассматривает и несколько процессорных ядер, и дополнительные аппаратные блоки системного уровня, часто называемые IP-ядрами (Intellectual Property Cores, IP Cores), которые присутствуют на микросхеме и могут порождать интерференцию. Таким образом, обобщенная озабоченность, поднятая в CAST-32A, – это угроза интерференции для некоторого программного компонента жесткого реального времени. В дополнение к озабоченности интерференцией CAST-32A определяет “робастное обособление”, которое дает измерения для демонстрации того, что обособление, наложенное на развертывание многоядерного процессора, робастно к угрозам интерференции как во времени, так и в пространстве.

Независимо от того, решит ли сегодняшний инженер переносить программный компонент жесткого реального времени в новый процессорный дом на массовой многоядерной микросхеме, угроза интерференции сроков для ПО реального времени является основной проблемой безопасности.

Если от массовых микросхем до 2005 года можно было ожидать физически изолированного одноядерного процессора, то колоссальные силы, обычно связываемые с законом Мура, увеличили и число аппаратных функциональных блоков на кристалле, и значимость угроз интерференции для сроков реального времени. Сегодня перенос процесса жесткого реального времени из прежнего дома на одноядерном процессоре в новый дом на системе на кристалле с многоядерным процессором требует большего внимания к интерференции, чем когда-либо ранее.

Проблемы интерференции в многоядерных процессорах (MCP)

CAST-32A организует вопросы, связанные с многоядерными процессорами, в виде набора вопросов. На эти вопросы можно ответить с учетом публикации FAA TC-16/51 2017 года. Измеримые ответы на вопросы CAST-32A позволят определить архитектуру обособления, достаточно “робастную” ко всем выявленным угрозам интерференции. “Темы, которые необходимо рассмотреть” в CAST-32A, то есть вопросы, сведены в следующей таблице:

Таблица 1: Сводка конкретных тем по многоядерным процессорам из CAST-32A

Код темы CAST-32A	Развернутое наименование темы	Что именно нужно рассмотреть или подтвердить
MCP_Planning_1	Планируемое применение многоядерного процессора (MCP)	Нужно определить, какой именно многоядерный процессор используется, сколько ядер и потоков активно, какие кэши, межсоединения, контроллеры памяти и иные особенности микросхемы участвуют в работе. Также требуется перечислить приложения, программные компоненты и компоненты архитектуры обособления, которые будут размещены на платформе.
MCP_Resource_Usage_1	Критичные значения конфигурации многоядерного процессора	Следует выявить и задокументировать настройки, которые влияют на робастное обособление ресурсов и времени: распределение ядер, настройки кэшей и памяти, параметры арбитража, режимы работы межсоединений, ограничения доступа и другие конфигурационные данные, от которых зависит детерминизм и безопасность.
MCP_Resource_Usage_2	Обнаружение и коррекция ошибок в критичных данных конфигурации (EDAC)	Нужно показать, как защищаются критичные данные конфигурации и связанные с ними данные состояния: за счет обнаружения ошибок, коррекции ошибок, контроля целостности или иных механизмов. Требуется подтвердить, что повреждение таких данных будет обнаружено, локализовано и не приведет к нарушению безопасности.
MCP_Planning_2	Перечень совместно используемых ресурсов	Требуется составить список всех совместно используемых ресурсов, например общей памяти, общей кэш-памяти, шин, межсоединений, контроллеров памяти, DMA, устройств ввода-вывода и других общих аппаратных блоков. Для каждого ресурса нужен план, как будет верифицироваться его использование, как будет предотвращаться исчерпание ресурса и как будет ограничиваться конкуренция между приложениями.
MCP_Resource_Usage_3	Перечень каналов интерференции	Нужно выявить каналы интерференции, через которые одна задача, ядро или аппаратный блок могут ухудшать поведение другого. Сюда входят конкуренция за память, кэш, межсоединения, периферию и иные общие ресурсы. Требуется определить, какие каналы реально применимы к целевой конфигурации, какие исключаются по обоснованию и какие меры снижения будут проверяться.
MCP_Resource_Usage_4	Использование ресурсов в наихудшем случае	Следует верифицировать, что потребность приложений в ресурсах многоядерного процессора и его межсоединениях не превышает доступную емкость даже в наихудшем случае. Эта проверка должна учитывать итоговую конфигурацию, выделение ресурсов, сценарии пиковых нагрузок и влияние конкурирующего доступа.
MCP_Software_1	Время выполнения в наихудшем случае для экземпляра программного компонента	Нужно показать, что каждая функция приложения выполняется корректно и успевает завершиться в пределах допустимого времени в итоговой конфигурации. По сути, требуется обоснование временной корректности приложения с учетом временного обособления, конкуренции за ресурсы и худших условий исполнения.
MCP_Software_2	Внутрикристальные ресурсы обмена данными и управления	Нужно верифицировать, что приложение корректно функционирует при всех настроенных связях по данным и управлению с другими программными компонентами, работающими на том же или на других ядрах. Особое внимание уделяется тому, что совместно используемые каналы связи и ресурсы внутри кристалла не нарушают робастное обособление и не создают недопустимую интерференцию.
MCP_Error_Handling_1	Отказы внутри многоядерного процессора и их влияние на цели безопасности	Следует выявить возможные внутренние отказы многоядерного процессора и показать, как они обнаруживаются, локализуются и обрабатываются. Нужно подтвердить, что такие отказы не приводят к недопустимому влиянию на другие обособленные компоненты и на цели безопасности системы либо что для этого предусмотрены эффективные меры снижения.
MCP_Acc_Summary_1	Дополненный итоговый отчет по ПО	Результаты рассмотрения всех перечисленных тем, принятые допущения, ограничения платформы, выполненные анализы, меры снижения и подтверждающие данные нужно свести в итоговый отчет по разработке ПО (Software Accomplishment Summary, SAS) или эквивалентный комплект итоговых данных.

Этот набор вопросов опирается на предварительное определение “ресурса”, которое в использовании CAST-32A зависит от “робастного обособления ресурсов”. Он также зависит от “робастного временного обособления” применительно к наихудшему времени выполнения (Worst-Case Execution Time, WCET) и включает несколько других определений, задающих рамки для корректных ответов на вопросы CAST-32A. Они обсуждаются ниже. Хотя CAST-32A не является официальной политикой или руководящим документом какого-либо сертифицирующего органа, его определения “робастного” обособления ресурсов и времени “следует обсуждать с соответствующим сертифицирующим органом” в ходе сертификационных проектов.

(Это “ПРИМЕЧАНИЕ” повторяется в нижнем колонтитуле каждой страницы CAST-32A.) Однако ясно и другое: никакого иного руководства по разработке системного ПО для многоядерных процессоров не существует, по крайней мере в форме, определенной и выраженной сертифицирующими органами. Поэтому CAST-32A – единственный жизнеспособный путь вперед.

Робастное обособление ресурсов

Что такое ресурс? Анализируя таблицу 1 (выше), мы обнаруживаем, что половина строк зависит от определения “ресурса” в том смысле, в каком оно используется в CAST-32A. Это вопросы MCP_Resource_Usage RU1, RU2, RU3, RU4 и перечень совместно используемых ресурсов Shared Resource List из P2. Все строки по меньшей мере косвенно зависят от определения “ресурса”: например, внутрикристальные коммуникации по данным и управлению из S2 тоже можно рассматривать как ресурсы.

Во-первых, отметим, что ресурс – это обособляемый ресурс. С точки зрения некоторого программного компонента жесткого реального времени, выполняющегося внутри многоядерного процессора, как CAST-32A определяет “робастное обособление ресурсов”? Следующим образом:

Робастное обособление ресурсов достигается, когда:

- обособленные компоненты ПО не могут загрязнять области хранения кода, ввода-вывода или данных других обособленных компонентов;
- обособленные компоненты ПО не могут потреблять больше выделенных им долей совместно используемых ресурсов;
- отказы аппаратуры, уникальной для обособленного компонента ПО, не могут вызывать неблагоприятные эффекты в других обособленных компонентах ПО.

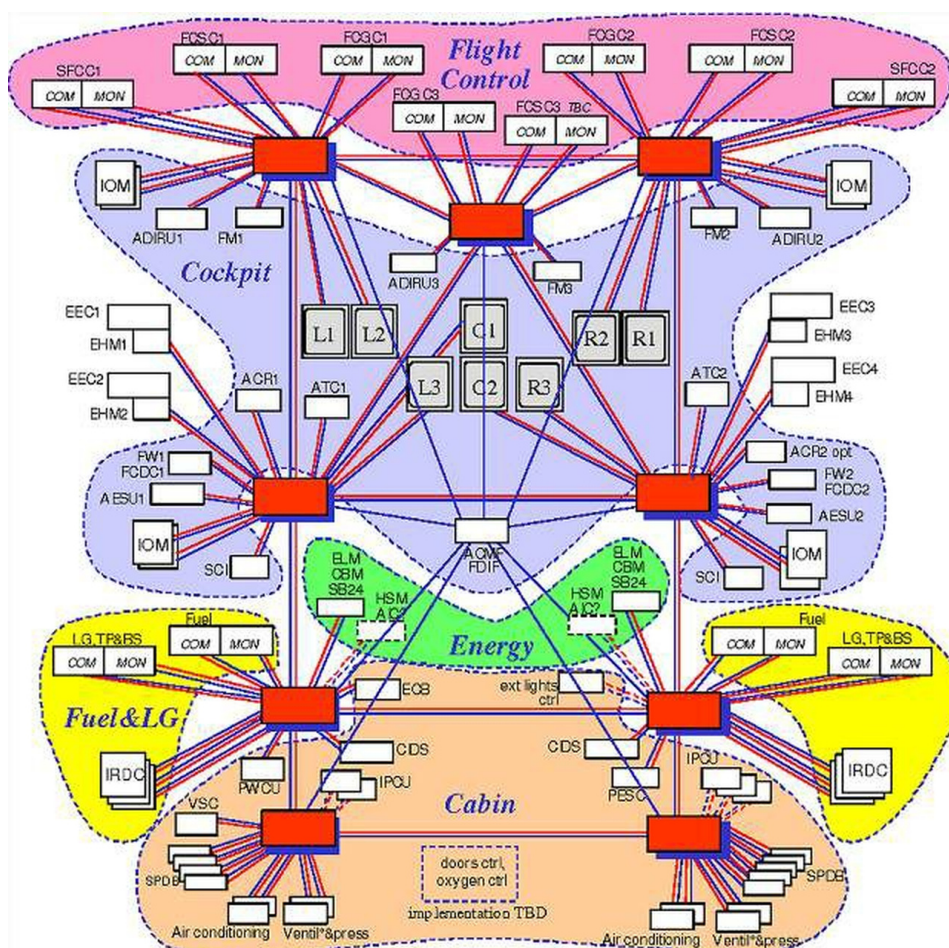
ПРИМЕЧАНИЕ: ПО, обеспечивающее обособление, должно иметь по меньшей мере тот же уровень гарантии разработки (Development Assurance Level, DAL), что и наивысший уровень гарантии разработки ПО, которое оно обособляет.

Это определение было “адаптировано из DO-248C / ED-94C и DO-297 / ED-124”. Иными словами, CAST-32A смог согласовать свое руководство по сертификации многоядерных процессоров с уже опубликованными сертификационными стандартами, рассматривающими архитектуру обособления. Если для ресурса уже задано обособление, DO-297 далее определяет ресурсы так, как описано в следующем разделе.

Обособление ресурсов: для интегрированной модульной авионики (ИМА) или для многоядерных процессоров (МСП)?

DO-297 определяет интегрированную модульную авионику (Integrated Modular Avionics, IMA) в абстрактном виде как “совместно используемый набор гибких, повторно используемых и взаимодействующих аппаратных и программных ресурсов, которые, будучи интегрированными, образуют платформу, предоставляющую сервисы” (приложение E, глоссарий), или, короче, как набор ресурсов. Конкретно ресурс определяется как “любой процессор, память, ПО, данные, объект или компонент, используемые процессором, платформой IMA, базовым ПО или приложением. Ресурс может совместно использоваться несколькими приложениями или быть выделенным для конкретного приложения. Ресурс может быть физическим (аппаратное устройство) или логическим (элемент информации)”. CAST-32A берет это абстрактное определение и адаптирует его к контексту многоядерных процессоров. В совокупности “робастное обособление” из DO-248C и “ресурс” из DO-297 образуют концепцию робастного обособления ресурсов в CAST-32A.

Заимствуя определение ресурса из DO-297, позиция CAST-32A предвосхищает сертификацию архитектуры IMA на многоядерных процессорах. В абстрактном смысле CAST-32A делает консервативный шаг, но для модульности, сети и замены последствия велики. В качестве примеров ресурсов IMA на одноядерных процессорах рассмотрим следующую схему из работы “The Airbus Approach to Open Integrated Modular Avionics (IMA)” (Н. Butz, 2007). Ресурсы доступны для ПО, выполняющегося внутри вычислителей, показанных белыми прямоугольниками:



Приведенная выше схема изображает “открытую IMA”, используемую на Airbus A380, и включает авиационный полнодуплексный коммутируемый Ethernet (Avionics Full-Duplex Switched Ethernet, AFDX), соединяющий множество одноплатных процессоров, стандартные авионические модули ARINC 600 (ARINC, Aeronautical Radio, Incorporated), модули базовой обработки и модули ввода-вывода. На схеме цветные области показывают логические домены использования IMA (Usage Domains, UD) самолета A380. В эти домены входят:

- управление полетом;
- кабина экипажа;
- энергоснабжение;
- топливо и шасси;
- салон.

Внутри этих областей белые прямоугольники показывают вычислительные модули на одноплатных процессорах, а линии обозначают подключение к сети AFDX (показаны девять резервированных коммутаторов Ethernet). Как обособленные ресурсы IMA включает временные интервалы Ethernet AFDX и датчики и исполнительные механизмы конкретных подсистем, а также общие вычислительные ресурсы, такие как память и временные интервалы процессора. Для обособления ресурсов, включая память и

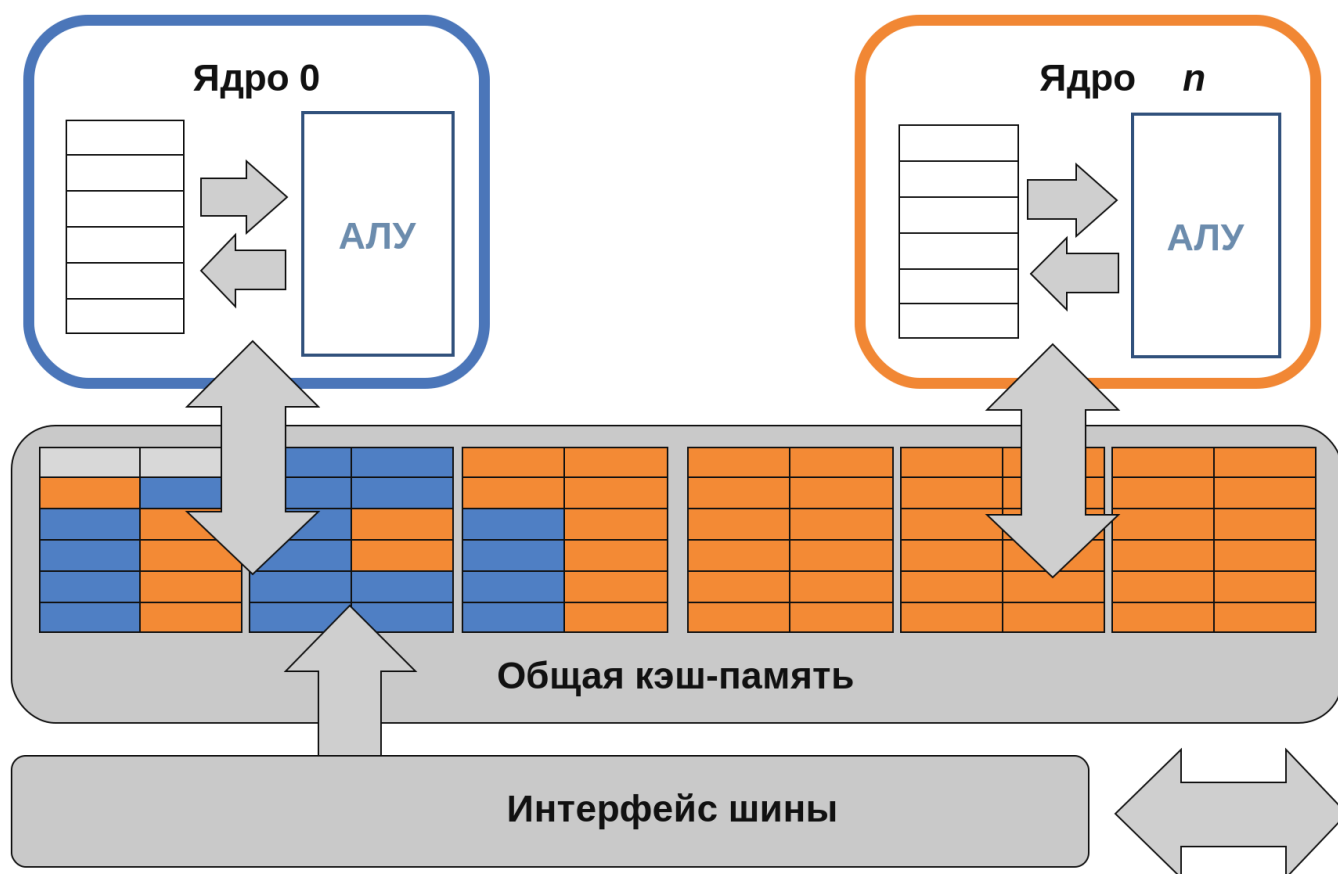
временные интервалы процессора, используется операционная система реального времени, OCPB (Real-Time Operating System, RTOS), обеспечивающая временное обособление и обособление устройств.

Интегрированная модульная авионика (ИМА) на кристалле

Хотя эта новая сеть ИМА должна обеспечивать коммуникации между обособленными компонентами внутри многоядерного процессора, ни одному обособленному компоненту не обязательно использовать коммутаторы Ethernet AFDX, позолоченные разъемы ARINC 600 или даже аппаратные блоки управления доступом к среде (Media Access Control, MAC) либо физического уровня (Physical Layer, PHY) Ethernet. Вместо этого простое совместное использование страниц динамической памяти между обособленными компонентами уже может удовлетворять мероприятию верификации емкости совместно используемых ресурсов из CAST-32A (MCP_Software_2).

С учетом относительной пропускной способности коммутируемого Ethernet по сравнению с доступом к внутрикристальной памяти интерфейс общей памяти может увеличить пропускную способность на несколько порядков. Но одновременно интерфейсы общей памяти могут вносить значительный недетерминизм, когда кэш-память используется совместно.

Кэш-память процессора существенно повышает производительность центрального процессора (Central Processing Unit, CPU) за счет временного хранения копий содержимого основной памяти, к которому, как предполагается, вскоре потребуется доступ. Это позволяет центральному процессору работать с часто используемыми данными без более медленного доступа к памяти через значительно более медленную шину памяти. Большинство процессоров имеют иерархию кэш-памяти разных уровней, например с первого по четвертый (Level 1 – Level 4, L1 – L4). Кэш-память нижних уровней обычно выделена конкретному ядру, а кэш-память более высоких уровней совместно используется процессорными ядрами для множественного доступа и еще большей пропускной способности. Это означает, что при использовании общей кэш-памяти центральный процессор должен управлять вопросами синхронизации процессора и кэша. Очевидно, что это значимая проблема интерференции ресурсов: состояние общей кэш-памяти может быть нарушено ошибочным или несинхронизированным использованием, что отрицательно скажется на производительности критически важного с точки зрения безопасности обособленного компонента. Например, любая крупная передача через интерфейс общей памяти, затрагиваемый кэш-памятью, особенно если эта кэш-память совместно используется с приложением или ядром, не участвующим в передаче, может вызвать существенное вытеснение данных из кэша при конкуренции с передачей по общей памяти. На схеме ниже показан один такой сценарий, влияющий на многоядерные процессоры:



Интерференция возникает, когда несвязанные приложения конкурируют за закрепление своего содержимого памяти в одной и той же общей кэш-памяти. Как показано выше, к многоядерному процессору добавлена общая кэш-память. Ядро 0 проигрывает ядру n конкуренцию за заполнение этой общей кэш-памяти, и в результате ядро 0 получает меньший детерминизм и большую задержку при доступе к адресам памяти из-за промахов кэша. Поэтому MCP_Software_2 может быть одновременно и ключевым прикладным сценарием, и ахиллесовой пятой вопросов многоядерных процессоров, поднятых в CAST-32A.

Робастное обособление и интерференция в многоядерных процессорах (MCP)

Соответствие CAST-32A требует ответа на следующий вопрос: можно ли в контексте IMA безопасно использовать многоядерный процессор для консолидации обособленных ресурсов из нескольких вычислительных модулей на одноядерных процессорах? Опираясь на множество ранее введенных определений, CAST-32A формулирует следующее определение подразумеваемого им робастного обособления:

“Платформа многоядерного процессора с робастным обособлением: платформа многоядерного процессора, которая соответствует целям настоящего документа и обеспечивает робастное обособление ресурсов и времени, как оно определено в настоящем документе, не только между программными приложениями, размещенными на

одном и том же ядре, но и между приложениями, размещенными на разных ядрах многоядерного процессора, или между приложениями, потоки которых размещены на нескольких ядрах”.

С учетом такого определения робастного обособления для платформы многоядерного процессора CAST-32A потребует учитывать любую поддерживаемую аппаратурой параллелизацию приложений. В частности, проектировщики должны явно указывать, каким приложениям какие аппаратные ядра или потоки выделяются, если это применимо. Рассмотрение этих вопросов потребует от разработчиков ПО учитывать следующие сценарии интерференции, представленные в порядке возрастания сложности:

Иерархия сценариев интерференции для многоядерных процессоров

1. Между приложениями на одном ядре.
2. Между приложениями на разных ядрах.
3. Между приложениями с потоками на нескольких ядрах.

Как следует из этой иерархии (выше), усилия по достижению робастного обособления можно упростить, минимизируя или устраняя интерфейсы и коммуникации между приложениями. Однако проблема возникает всякий раз, когда приложение или выделенный приложению временной интервал мигрирует с одного аппаратного ядра (или потока) на другое. Поскольку операционные системы SMP рассматривают все ядра и все временные интервалы ядер, с точки зрения диспетчеризации, как симметрично взаимозаменяемые, побочным следствием этого соглашения SMP становится то, что сведения о выделении приложения тому или иному ядру скрываются операционной системой от разработчика ПО. По той же логике операционная система SMP, как правило, свободна перенести приложение на новое ядро после вытеснения.

Хотя разработчик авиационного ПО может стремиться минимизировать случаи, когда потоки приложения распределяются по нескольким ядрам (см. уровень 3 выше), операционная система SMP может не предоставлять хороших средств, чтобы навязать асимметричные различия ядрам, которые считаются симметричными. В крайнем варианте в некоторых авиационных конструкциях отключают все ядра многоядерного процессора, кроме одного. Это гарантирует привязку приложения к ядру и при этом сохраняет многие преимущества операционной системы. Но главное преимущество проектов на многоядерных процессорах состоит в использовании больших вычислительных возможностей одновременной обработки и повышенной вычислительной мощности, а оно теряется при отключении всех ядер, кроме одного. Поэтому более удовлетворительные решения для многоядерных процессоров будут включать как гарантированное выделение приложений ядрам, так и обеспечение робастного обособления в смысле определения CAST-32A.

Проследив несколько определений назад до DO-297 (2005), CAST-32A снимает с себя необходимость заново определять архитектуру обособления, ее обособленные ресурсы, а также цели конфигурирования, интеграции и другие цели IMA. Вместо этого CAST-32A ставит свои вопросы именно о платформах многоядерных процессоров, исходя из того, что “робастное обособление” уже определено.

Таблица 2: Предлагаемая интерпретация тем CAST-32A по фазам жизненного цикла

Фаза жизненного цикла	Тема CAST-32A для рассмотрения	Предлагаемая интерпретация
Планирование	MCP_Planning_1	Перечень ресурсов: идентифицировать микросхему многоядерного процессора, число активных ядер, их тактовые частоты, их кэши и любые динамические особенности, влияющие на процессорные ядра. Перечислить приложения и компоненты, составляющие архитектуру обособления, включая любые используемые компоненты OCPB, ARINC 653 и/или гипервизора.
Планирование	MCP_Planning_2	Перечень совместно используемых ресурсов: документировать план, включающий: • перечень совместно используемых ресурсов, включая планы того, как верифицировать их использование, как избежать или снизить конкуренцию и как предотвратить их исчерпание из-за требований, предъявляемых к ним запланированными программными или аппаратными компонентами; • перечень динамических особенностей аппаратуры, которые будут активны, и план того, как они будут использоваться.
Планирование	MCP_Resource_Usage_1	Конфигурация обособления: документировать данные настроек конфигурации, которые управляют обособлением ресурсов и времени, выявленным в P1 и P2.
Планирование	MCP_Resource_Usage_3	Анализ интерференции. Идентифицировать каналы интерференции, которые могут позволить интерференции влиять на программные приложения, размещенные на ядрах многоядерного процессора. • Учесть совместно используемую память, кэш, межсоединения или аппаратные ресурсы ввода-вывода. • Описать подход к верификации, запланированный для любых мер снижения, выбранных заявителем. ◦ Исключить каналы интерференции, которые не могут влиять на программные приложения при предполагаемой итоговой конфигурации системы. ◦ Включить любые каналы интерференции, обнаруженные в любой момент в ходе проекта. ◦ В некоторых случаях IDAL С исключить эту цель и соответствующий анализ.
Сборка и повторное использование	MCP_Resource_Usage_2	Обнаружение и коррекция ошибок (Error Detection and Correction, EDAC) конфигурации: разработать, документировать и верифицировать средства защиты данных критических настроек конфигурации для многоядерного процессора, включая EDAC.
Сборка и повторное использование	MCP_Error_Handling_1	Анализ отказов и снижение последствий. Идентифицировать, спланировать, спроектировать, реализовать и верифицировать средства обнаружения и обработки отказов, которые могут возникать внутри многоядерного процессора. Локализовать влияние любых отказов в пределах оборудования, в котором установлен многоядерный процессор; для этого может использоваться "сеть безопасности" внутри этого оборудования.
Верификация	MCP_Resource_Usage_4	Емкость ресурсов. Верифицировать, что требования к ресурсам многоядерного процессора и межсоединениям не превышают емкость с учетом обособления по P1, предполагаемых итоговых настроек конфигурации и идентификации сценариев наихудшего случая.
Верификация	MCP_Software_1	Наихудшее время выполнения приложения. Верифицировать, что все функции приложения работают корректно и имеют достаточно времени для завершения выполнения при предполагаемой итоговой конфигурации, включая ее временное обособление.
Верификация	MCP_Software_2	Емкость совместно используемых ресурсов. Верифицировать корректное функционирование каждого приложения с учетом всех сконфигурированных связей по данным и управлению между ним и всеми подключенными программными компонентами, выполняющимися на том же или на других ядрах многоядерного процессора.
Отчетность	MCP_Acc_Summary_1	Итоговый отчет: добавить ответы на эти вопросы в итоговый отчет по ПО (Software Accomplishment Summary, SAS) по DO-178C.

Понимание того, как CAST-32A заимствует концепции обособления ресурсов из контекста IMA, облегчает интерпретацию CAST-32A. Реалистичный вопрос состоял бы в том, как новая многоядерная микросхема может консолидировать множество вычислительных модулей IMA предыдущего поколения на одноядерных процессорах,

например вставные модули ARINC 600, чтобы дополнительно сократить габариты, массу, энергопотребление и стоимость (SWaP-C). В этом контексте вопросы CAST-32A можно интерпретировать следующим образом (см. схему на следующей странице).

При такой интерпретации “темы, которые необходимо рассмотреть” из CAST-32A распределяются по фазам жизненного цикла разработки системы на основе анализа описанных в CAST-32A усилий, требуемых для задачи. Цель состоит не в том, чтобы изменить задачу, как она сформулирована в CAST-32A, а в том, чтобы помочь оценить, на каких этапах жизненного цикла разработки системы эти задачи должны выполняться. Кроме того, предлагаемая интерпретация указывает цель, то есть то, на что инженер и руководитель должны ориентироваться для завершения задачи. Исходя из этой интерпретации, большинство задач CAST-32A должно быть включено как новые мероприятия фазы планирования, а также как мероприятия верификации после анализа покрытия.

Проблемы анализа интерференции по CAST-32A

CAST-32A можно интерпретировать как документ, организованный по фазам жизненного цикла: он выделяет новые задачи планирования, реализации и интеграции-верификации (см. таблицу 2 выше), тем самым напоминая DO-178C. Инженеры и руководители, которые намерены сертифицировать одно или несколько приложений по CAST-32A для многоядерных процессоров, должны выделить работы и персонал для решения этих задач. При этом основные усилия, как можно ожидать, возникнут после завершения верификации на основе покрытия, когда может начаться тестирование на основе производительности. Напомним: “покрытие” сначала означает “покрытие требований”, а для уровня гарантии разработки C и выше к нему добавляется “структурное покрытие”. Исследование, спонсированное FAA и изложенное в TC-16-51, использовалось для совершенствования CAST-32 до CAST-32A и теперь служит информативным дополнением к CAST-32A. Оно определяет эти усилия следующим образом:

“Анализ интерференции в первую очередь является вопросом оценки производительности. Если предположить, что в идеале у процессора нет режимов отказа, запускаемых каналами интерференции, то некоторые каналы интерференции будут иметь ограниченное и приемлемое влияние на время выполнения ПО. ...” (TC-16-51, с. 11)

Хотя CAST-32A в общем виде предполагает проектные усилия по выполнению анализа интерференции, FAA TC-16-51 намного подробнее объясняет, что именно нужно анализировать, чтобы эти предполагаемые усилия были успешными. Если традиционно для обеспечения безопасности полетов в авиационной отрасли требовались нисходящие анализы опасностей и режимов отказа, то TC-16-51 рекомендует: после того как нисходящий анализ распределил приложения IMA по обособленным компонентам,

называемым доменами использования (Usage Domains, UD), как в открытой IMA Airbus (см. выше), должен быть также выполнен восходящий анализ интерференции. В частности, TC-16-51 утверждает:

Применение нисходящего подхода является необходимым шагом для установления границ исследования источников недетерминизма [интерференции] в рамках восходящего подхода (обсуждается в разделе 6.1.2). Хотя [нисходящий подход] облегчает промышленные и сертификационные процессы... этого подхода недостаточно. ... Нисходящий подход задает критерии выбора для многоядерной микросхемы, а также для выбора IP-ядер, которые будут активированы или сконфигурированы в составе решения на многоядерной микросхеме. Эта конфигурация определяет домен использования многоядерной микросхемы, в котором [должен быть] гарантирован детерминизм. (TC-16-51, с. 29, выделение добавлено).

Результаты нового восходящего подхода, предложенного TC-16-51, можно суммировать как ответы на вопросы CAST-32A, а затем свести к ответам инженера по MCP_Software_2. Анализируя интерпретацию CAST-32A в таблице 2, авторы полагают, что полный набор вопросов, поднятых CAST-32A, создаст систему измерений, полезную для наблюдения релевантных каналов интерференции. Измеренные результаты интерференции, сначала для наихудшего времени выполнения по MCP_Software_1 с учетом выделенных ресурсов, а затем для отсутствия интерференции по MCP_Software_2 с учетом как выделенных, так и совместно используемых ресурсов, будут агрегированы в итоговую оценку интерференции на многоядерном процессоре, измеренной снизу вверх.

Описание методов и анализов восходящего измерения для различных аппаратных ресурсов, представленных в TC-16-51 и его ссылках, выходит за рамки этой статьи. На уровне обсуждения CAST-32A важно другое: восходящее измерение аппаратно обусловленных каналов интерференции может быть классифицировано следующим образом, а затем суммировано для ресурса в каждом обособленном компоненте домена использования:

"Метки" анализа интерференции, применяемые к наблюдаемым аппаратным каналам

Приемлемо

- **Ограниченный:** канал интерференции имеет ограниченное воздействие, поэтому можно оценить штраф интерференции.
- **Известно допустимый:** измеренный штраф соответствует требованиям функционального домена оборудования.

Неприемлемо

- **Ограниченный:** канал интерференции имеет ограниченное воздействие.
- **Известно недопустимый:** оцененный штраф интерференции слишком велик.
- **Требуется смягчение:** необходимо внести изменения для выполнения требований к характеристикам.

Неограничено

- **Неограниченный:** для этого канала невозможно оценить штраф интерференции.
- **Неизвестно недопустимый:** воздействие на время выполнения ПО надлежащим образом не известно.
- **Требуется смягчение:** необходимо внести изменения для обеспечения требований к характеристикам.

Отказ

- **Неограниченный с отказом:** канал интерференции вызвал режим отказа в процессоре или многоядерном процессоре.
- **Известно недопустимый:** требуется дальнейшее исследование совместно с изготовителем и формализация через анализ безопасности.
- **Запрошено смягчение:** у изготовителя запрошено устранение.

Насколько трудно выполнить этот анализ интерференции для всех каналов интерференции и для всех ресурсов в домене использования ИМА? Формула (ТС-16-51, с. 56) и таблица, демонстрирующая экспоненциальный рост числа путей интерференции на примерах, показывают, что такая задача недостижима, кроме очень простых архитектур (ТС-16-51, с. 57). ТС-16-51 рекомендует то, что можно интерпретировать как все “возможные упрощения в стратегиях обеспечения детерминизма на уровне многоядерного процессора” (с. 30). Без таких стратегий выполнение анализа может оказаться трудным или невозможным. “Основная проблема проведения анализа интерференции состоит в достижении компромисса (по времени и стоимости) между испытательными кампаниями и необходимостью иметь заслуживающее доверия покрытие ситуаций интерференции” (ТС-16-51, с. 11). Кратко говоря, выбор очень консервативной архитектуры обособления

может быть наиболее успешным подходом к сертификации системы на многоядерном процессоре по CAST-32A в том виде, в каком CAST-32A сейчас уточняется документом TC-16-51.

Преимущества использования интегрированной модульной авионики (ИМА)

Инженеры и руководители, которые намерены сертифицировать одно или несколько приложений по CAST-32A для многоядерных процессоров, должны учитывать влияние на жизненный цикл разработки, сходное с другими проектами на основе ИМА. По существу, за последующую выгоду нужно заплатить начальную цену: “бесплатных обедов не бывает”. Однако последующий обед, хотя и не будет бесплатным, должен обойтись значительно дешевле: преимущества лучшей интеграции ИМА, повторного использования и качества должны перевесить дополнительные первоначальные затраты. Один из способов оценить преимущества ИМА – рассмотреть, что может быть сертифицировано по консультативному циркуляру FAA AC 20-170. Названия глав показывают, что может быть сертифицировано и одобрено по правилам FAA для ИМА, в отличие от сертификации только систем:

- Глава 2. Принятие и поэтапное принятие
- Глава 3. Получение письма о принятии компонента ИМА
- Глава 4. Повторное использование компонентов ИМА
- Глава 5. Управление конфигурацией в системе ИМА
- Глава 6. Функции восстановления ИМА

Преимущества ИМА видны уже из этого фрагмента оглавления AC-20-170. Они включают сокращение сертификационных усилий за счет поэтапного принятия, принятия компонентов и повторного использования уже принятых компонентов. Эти экономические выгоды могут накапливаться годами, на уровне нескольких подсистем, компонентов, поставщиков и/или систем воздушного судна.

Соображения по использованию интегрированной модульной авионики (ИМА) на многоядерных процессорах

Поскольку определения модулей, компонентов и приложений ИМА включают и аппаратуру, и ПО, после сертификации они могут пользоваться преимуществами модульной замены на аппаратной основе. Это выгодно отличается от практики сертификации и одобрения FAA для чисто программных подходов к модульности. Практическое следствие использования аппаратной платформы для модулей ИМА состоит в том, что модульность любой конкретно реализованной системы ИМА сводится к выбранному ею стандарту подключаемости. Например, в открытой ИМА Airbus стандартным соединителем является позолоченный штекер ARINC 600. С учетом этих практических последствий правила сертификации и принятия ИМА из DO-297 и AC 20-170 можно интерпретировать как подчеркивающие аппаратные стандарты

подключения, например сеть Ethernet AFDX или штекеры ARINC 600. Однако для архитектур IMA использовались как AFDX (ARINC 664), так и MIL-STD-1553 и ARINC 429, а исследование Национального управления по аэронавтике и исследованию космического пространства (National Aeronautics and Space Administration, NASA) по сравнению коммуникационных архитектур для аэрокосмических модульных авиационных систем рассматривает различные коммуникационные протоколы, которые могут использоваться в коммуникационных системах IMA (NASA/TM 2006-214431).

Будущие поколения архитектур IMA на базе многоядерных процессоров должны учитывать эти практические соображения. Стандарты связности IMA должны измениться, когда приложения IMA переносятся на многоядерные процессоры, потому что робастно обособленные компоненты в CAST-32A – это уже не сетевые вычислительные модули с возможностью линейной замены, а скорее их эквиваленты в виде виртуальных машин: например, одно ядро или временной интервал ядра внутри многоядерного процессора и его обособленные ресурсы. Как один временной интервал ядра обменивается данными с другим временным интервалом на том же многоядерном процессоре? Или, что, возможно, важнее, как сохранить полезную подключаемость компонентов IMA, когда физические штекеры ARINC 600 поглощаются кристаллом многоядерного процессора? Хотя специфичные для реализации стандарты могут устареть при переносе компонентов IMA на многоядерные процессоры, потребность в стандартизованных коммуникациях и подключаемости, поддерживающей линейную замену, останется полностью неизменной.

Влияние CAST-32A

Для инженеров и руководителей CAST-32A и поддерживающий его TC-16-51 открывают дверь к будущим многоядерным микросхемам и микросхемам с очень большим числом ядер, которые уже появляются на горизонте прогнозов закона Мура. Ожидается, что традиционные авиационные приложения “переедут” и найдут новый дом на многоядерных процессорах.

Для более широкой отрасли авионики позиция CAST-32A может направить сырую вычислительную мощность на полезную работу. Успехи коммерциализации готовых коммерческих микросхем (Commercial Off-The-Shelf, COTS) усиливают потребность в авиационных многоядерных процессорах, как и предполагалось в CAST-32A. Однако из-за сложных вопросов, поставленных в CAST-32A, и трудности избегания рисков от источников недетерминизма в многоядерных процессорах, как показано в TC-16-51, текущий консенсус FAA, по-видимому, создает метафорическую паровую машину, где поршнем является робастное обособление интегрированной модульной авионики.

Вопросы “Каковы архитектурные стратегии?” (раздел 2.4) и “Что такое архитектура обособления?” (раздел 2.4.1), уже присутствующие в DO-178C, теперь можно рассматривать как сходящиеся в CAST-32A, включая его опору на IMA. С учетом исторических успехов IMA в военной и гражданской авиации, а также в космических системах NASA, использование определений IMA из DO-297 в CAST-32A для поддержки

робастного обособления представляется оправданным. Если будущая отрасль авионики и аэронавтики сойдется на платформах многоядерных процессоров для IMA, CAST-32A будут рассматривать как переломный момент.

Затраты и выгоды авиационной разработки и сертификации

Автор: Вэнс Хилдерман

Рецензент:

г-н Кармело Томмази

Введение.

“DO-178, 254, 278 – худшие стандарты в мире... за исключением всех остальных.”
(Вэнс Хилдерман, 2004)

Эта переработка знаменитой фразы Уинстона Черчилля о демократии содержит долю истины: руководства серии DO-XXX могут быть и благом, и проклятием для авиационных проектов по всему миру. Многие пользователи жаловались на дополнительные затраты, связанные с DO-178, с момента появления этого документа сорок лет назад, а последующие версии и родственные руководства не снизили стоимость, потому что добавили еще больше строгости. DO-XXX и ARP47XX действительно никогда не бывают дешевыми, особенно в первом проекте. В явных случаях, рассмотренных ниже, при первой попытке DO-XXX и ARP47XX увеличивают затраты минимум на 20-40 % сверх того, что в иных условиях потратили бы сильные инженерные компании. Но действительно ли DO-XXX и ARP47XX “слишком” дороги? Не снижают ли они долгосрочные затраты для компаний, которые и так вели разработку разумно? Снижают ли они долгосрочные затраты ценой большей начальной стоимости разработки? Повышают ли они безопасность полетов и надежность, и если да, то насколько? Какие именно выгоды дает соответствие DO-XXX и ARP47XX? Могут ли компании снизить затраты на сертификацию без потери качества? Эти важные вопросы рассматриваются ниже.

DO-XXX и ARP47XX постепенно стали стандартами де-факто почти для всех видов гражданской авиации, кроме экспериментальных воздушных судов. За последнее десятилетие DO-178B, а теперь DO-178C/DO-254, стали обязательными для большей части военной авионики. Новый DO-178C и меморандумы Федерального управления гражданской авиации США (FAA) и Агентства Европейского союза по авиационной безопасности (EASA), актуализирующие DO-254, увеличивают затраты для тех пользователей, которые обходились упрощениями при применении прежнего DO-178B и раннего DO-254. Однако эти руководства обладают признаками, общими для всех областей, критически важных с точки зрения безопасности: планирование, согласованность, детерминизм, тщательная документация и тестирование, а также

доказательство всех перечисленных свойств. DO-XXX/ARP47XX в значительной мере опираются на верификацию при оценке качества авиационного изделия. Однако настоящее качество появляется благодаря качественному проектированию и процессу реализации, а не благодаря тестированию.

Эти авиационные “стандарты” исходят из того, что воздушное судно, системы, аппаратура и ПО должны работать согласованно, а надежность каждого элемента должна быть доказана. Они образуют общую экосистему, включающую руководства по безопасности полетов, системам и смежным вопросам, как показано ниже:



Базовая экосистема авиационной разработки и сертификации

До появления DO-254 аппаратура считалась “видимой” и испытывалась на системном уровне вместе с интегрированным ПО, поэтому на нее не распространялись цели DO-178B. Но это исключение привело к переносу функциональности из ПО в аппаратуру ради обхода сертификации ПО. Кроме того, сложность аппаратуры выросла настолько, что из-за встроенной логики в программируемых логических устройствах (Programmable Logic Devices, PLD), специализированных интегральных схемах (Application-Specific

Integrated Circuits, ASIC) и ПЛИС (Field-Programmable Gate Arrays, FPGA) она часто не уступает сложности ПО. Теперь все признают, что аппаратура и ПО образуют неразрывную цепь, качество которой определяется самым слабым звеном. Отсюда и требование применять DO-254 также к авиационной аппаратуре: по мере развития сертификации авионики слабые звенья последовательно устраняются. Руководство по разработке воздушных судов и систем (ARP4754A) требует явно учитывать всю эту систему на протяжении жизненного цикла авиационной разработки, поэтому DO-178C/DO-254 согласованы с ARP4754A.

В DO-178/278 под ПО понимаются все драйверы, пакеты поддержки плат (Board Support Packages, BSP), операционные системы, ОС (Operating Systems, OS), операционные системы реального времени, OCPB (Real-Time Operating Systems, RTOS), библиотеки, графика и прикладное ПО, то есть любая исполняемая логика или данные, которые загружаются в память или используются при выполнении в реальном времени. Тестирование означает подтверждение того, что самые детальные требования низкого уровня корректно реализованы, пути выполнения покрыты в соответствии с их уровнем критичности, а полная двунаправленная трассируемость обеспечена. Для автоматизации процессов DO-XXX и ARP47XX все шире применяются инструменты. Например, почти все соответствующие требованиям авиационные проекты сегодня используют коммерческий инструмент трассируемости для управления связями между требованиями, реализацией и тестами (см. главу о трассируемости в этой книге). Хотя такие инструменты формально не обязательны, практически каждый соответствующий требованиям проект использует их, чтобы выполнить требования DO-XXX/ARP47XX к трассируемости сверху вниз и снизу вверх. Формального требования к таким инструментам нет (трассируемость всегда можно вести вручную), но инструмент трассируемости, соответствующий DO-XXX/ARP47XX, существенно снижает стоимость достижения соответствия почти для всех проектов, кроме самых простых. Сегодня то же справедливо для инструментов управления конфигурацией (Configuration Management, CM), автоматизированного тестирования, анализа структурного покрытия и статического анализа логики: многие такие инструменты вышли на рынок специально под потребности соответствия DO-XXX и ARP47XX. Как уже отмечалось ранее, стоимость приобретения и кривая обучения для этих средств инженерной автоматизации означают, что первая проектная команда, применяющая их, скорее всего, не получит экономического выигрыша. То же относится к созданию повторно используемых программных и аппаратурных компонентов. Зато последующие проекты получают большую выгоду от этих инструментов и лучшего повторного использования. Именно здесь DO-XXX/ARP47XX проявляют себя особенно убедительно.

Затраты на авионику в авиационной отрасли

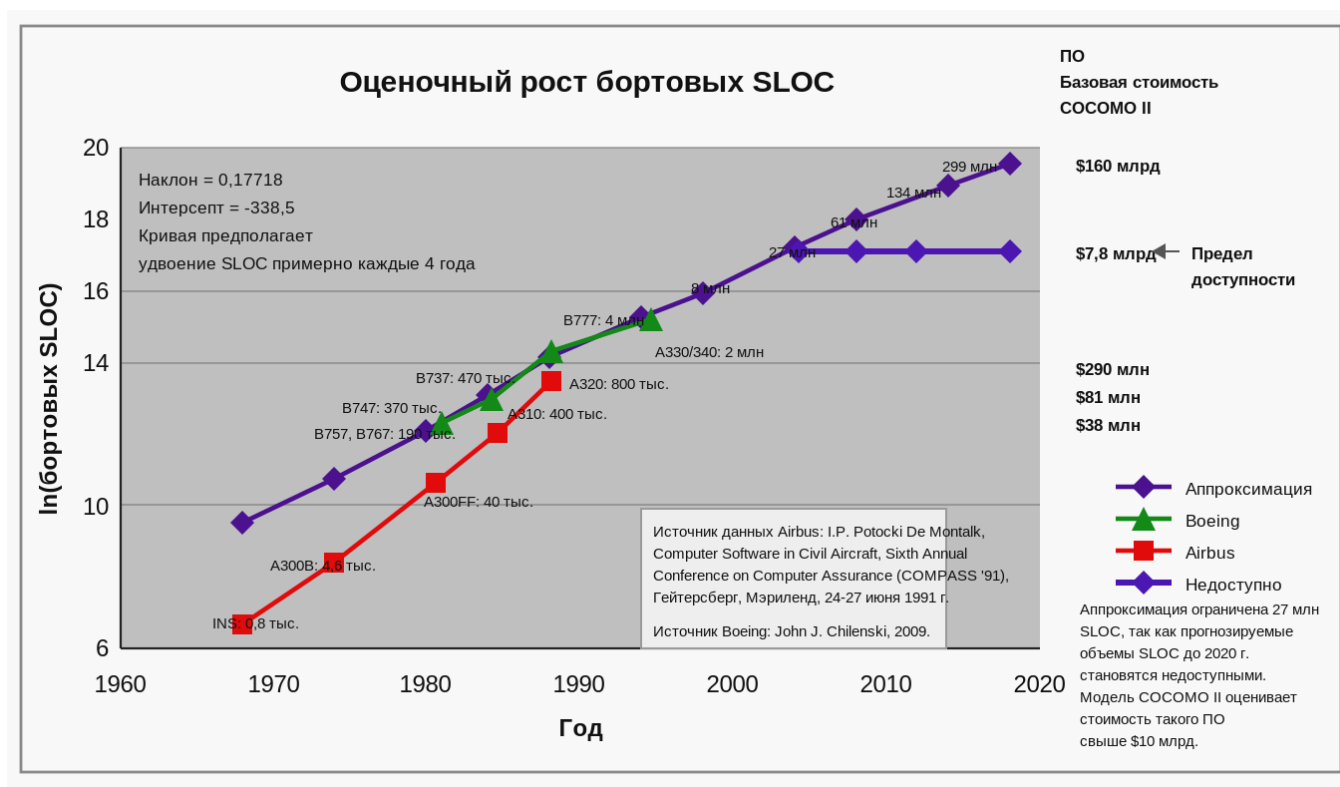
Системы, критически важные с точки зрения безопасности, заметно дороже в разработке, чем потребительские устройства и коммерческие системы информационных технологий, ИТ (Information Technology, IT). По сути, повышенная отказоустойчивость и

более высокая надежность имеют существенную цену. По мере роста сложности авиационных систем инженерные затраты на научно-исследовательские и опытно-конструкторские работы, НИОКР (Research and Development, R&D), связанные с этой сложностью, также растут экспоненциально. Приведенная ниже таблица взята из финансируемого исследования, выполненного крупной консультационной компанией в области авиационной разработки. Погрешность данных составляет 20 %, поскольку часть предположений и данных невозможно было подтвердить.

Тип воздушного судна	Приблизительное общее число строк ПО и логики	Средняя стоимость одной строки ПО и логики	Стоимость разработки ПО
Поршневые воздушные суда авиации общего назначения, 6+ мест	100 тыс. – 1,5 млн	\$260	\$26 млн – \$390 млн
Gulfstream G550	4,0 млн	\$330	\$1,32 млрд
Boeing 737 NG (600) (1995)	3,5 млн	\$240	\$700 млн
Boeing 777 (ранняя версия – 995)	5,5 млн	\$260	\$1,04 млрд
Boeing 787 (2011)	7 млн	\$370	\$2,59 млрд
JSF-35 (единственный ударный истребитель; Joint Strike Fighter, JSF; по состоянию на 2019 г.)	15 млн	\$520	\$7,8 млрд (бортовая часть)

Сводка затрат на программную инженерию воздушного судна по типам воздушных судов

В отдельном исследовании Институт программной инженерии (Software Engineering Institute, SEI) Университета Карнеги-Меллона сравнивает связанный набор данных по строкам исходного кода (Source Lines of Code, SLOC) ПО Airbus и Boeing. Эти данные показаны на рисунке ниже. Обратите внимание: SEI утверждает, что число строк исходного кода будет удваиваться каждые четыре года, и предполагает больший объем кода на тип воздушного судна, чем в предыдущем исследовании на предшествующем рисунке:



Источник: Институт программной инженерии, Университет Карнеги-Меллона

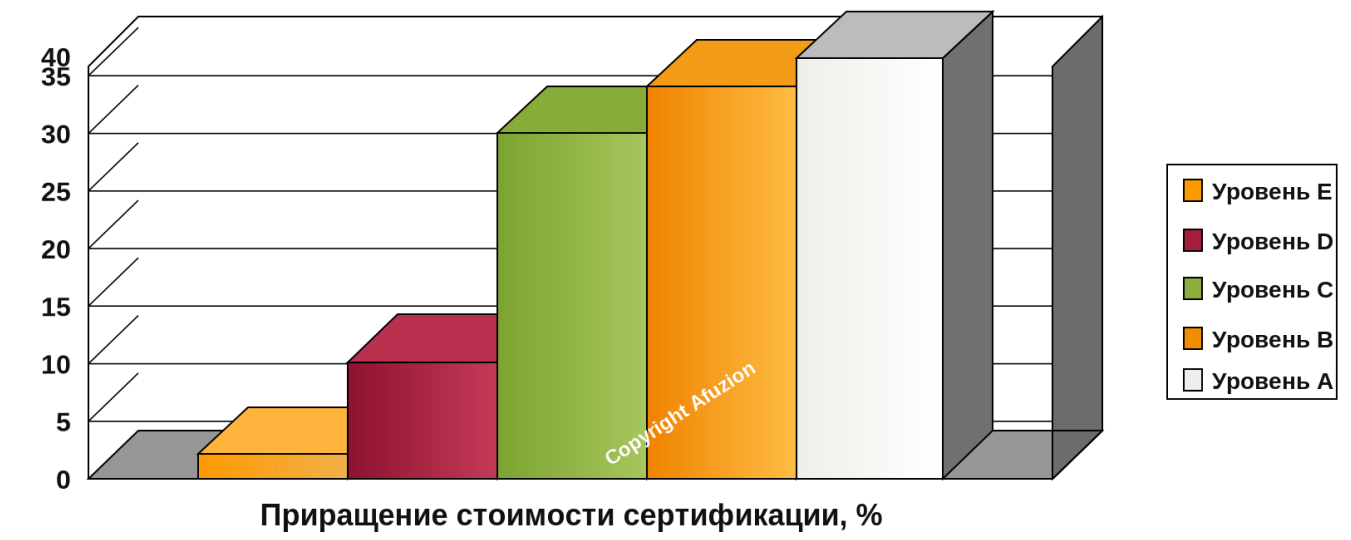
Интересно, что космическая программа “Аполлон” добилась больших успехов в разработке, хотя почти не имела стандартов разработки. Однако сложность ее систем и объем логики составляли менее 1 % от современных коммерческих воздушных судов, а требования к сертифицируемости или повторному использованию отсутствовали полностью. В действительности современные воздушные суда в значительной мере являются развитием ранее созданных систем, поэтому повторное использование этих систем имеет первостепенное значение. Именно здесь DO-XXX/ARP47XX существенно помогают увеличить повторное использование и сократить время интеграции, как подробно описано ниже.

DO-XXX/ARP47XX: уровни критичности и стоимость.

DO-178 и DO-254 предусматривают пять уровней критичности: от уровня А (наиболее критичный) до уровня Е (наименее критичный). В DO-200 таких уровней три, а в DO-278 – шесть; причины этого уже объяснялись в соответствующих главах книги. Каждой авиационной системе по результатам оценки безопасности назначается один или несколько уровней критичности, и каждый элемент этой системы должен соответствовать назначенному ему уровню критичности или превосходить его. С ростом критичности растет и строгость требований к документации, проектированию, рассмотрению, реализации, верификации, гарантии качества, гарантии процесса и независимости. Соответственно растут стоимость и сроки. Следующий график и связанная с ним таблица показывают фактические приращения стоимости для каждого уровня критичности. В качестве базы дополнительные затраты на DO-178С сопоставлены с разработкой ПО среднего качества, например качественного коммерческого или потребительского ПО, где

фактически применимый уровень зрелости разработки по интеграции моделей зрелости возможностей (Capability Maturity Model Integration, CMMI) находился между уровнями 3 и 4.

График: приращение стоимости и сроков по DO-178 в зависимости от уровня критичности сверх CMMI 3



Уровень Е Стоимость	Уровень D Стоимость	Уровень С Стоимость	Уровень В Стоимость	Уровень А Стоимость
Базовый уровень	E+15 %	D+35 %	C+10 %	B+5 %

Таблица: приращение стоимости и сроков по DO-178C для каждого уровня критичности относительно базовой стоимости

Распространенный миф состоит в том, что DO-XXX/ARP47XX чрезмерно дороги и увеличивают затраты более чем на 200 %. Да, они не “дешевые”, что видно по приведенным выше начальным дополнительным затратам. Сертифицированные системы уровня D обычно все же имеют полноценное планирование, хорошие требования с трассируемостью к тестам, документированную реализацию, рассмотрения и полное функциональное тестирование всех требований с применением трассируемости. Кроме того, даже на уровне D применяются управление конфигурацией, гарантия качества и взаимодействие с сертифицирующим органом. Как говорилось в предыдущей главе о DO-178, у уровня D есть 26 целей, то есть на 26 больше, чем у вашего смартфона или любительского беспилотника. При этом затраты на уровень D всего на 15 % выше, чем затраты на ПО среднего качества для потребительского/коммерческого рынка, разработанное по типичным для отрасли процессам CMMI уровня 2-3. Почему? Потому что уровень D почти полностью состоит из обычных отраслевых инженерных принципов: планы, требования, тесты, трассируемость, рассмотрения и базовые процессы управления конфигурацией и гарантии качества. Кроме того, многие компании, впервые работающие с DO-XXX/ARP47XX, считают, что их прежние работы, включая планирование, требования, проектирование, тестирование и рассмотрения, придется выполнять заново.

Это неверно. Хорошо проведенный анализ пробелов (Gap Analysis) позволяет оценить существующие процессы и работы, повторно использовать пригодные результаты и выявить пробелы, которые нужно закрыть для выполнения требований DO-XXX/ARP47XX.

DO-XXX/ARP47XX также требуют рассмотрений всех процессов и артефактов разработки ПО; для уровней А и В такие рассмотрения в основном должны быть независимыми. Рассмотрения выполняются по утвержденным контрольным перечням, чтобы все требуемые аспекты действительно были проверены. Чтобы экономить деньги и время, многие компании используют автоматизированные инструменты управления проектом и рассмотрениями; поставщиков таких инструментов легко найти через обычный интернет-поиск. Пример такого контрольного перечня приведен в приложении к этой книге.

Еще один миф – будто самый большой рост стоимости ПО происходит при переходе с уровня критичности В на уровень А. Это неверно, и причина довольно интересна. Самое крупное отличие системы уровня А от системы уровня В состоит в том, что по ARP4761A для системы уровня А требуется надежность в 100 раз выше. Но эта надежность должна обеспечиваться прежде всего архитектурой системы/аппаратуры, а не ПО. За счет чего? За счет дополнительного резервирования: обычно единственный способ получить надежность уровня А – увеличить резервирование аппаратуры/системы, что, конечно, сильно повышает общую стоимость изделия. А вот разница в стоимости ПО между уровнями А и В, как видно из приведенных выше данных, весьма мала (+5 %).

Наиболее существенная разница в стоимости внутри DO-178C наблюдается между уровнями D и C. То же справедливо для систем связи, навигации, наблюдения и организации воздушного движения (CNS/ATM, Communication, Navigation, Surveillance / Air Traffic Management), разрабатываемых по DO-278A, между уровнями гарантии 3 и 4. Почему? Уровень C требует следующих ключевых целей, которых нет на уровне D; именно они приводят к тому, что уровень C требует на 35 % больше усилий, чем уровень D:

- тестирование требований низкого уровня к ПО;
- обеспечение 100 %-ного покрытия всех операторов исходного кода;
- оценка требований, проекта и кода на соответствие стандартам;
- более строгие рассмотрения;
- во многих случаях – более строгий режим управления конфигурацией.

Уровень В требует дополнительного структурного покрытия (покрытия решений и условий, то есть всех ветвей в исходном коде), дополнительной независимости при рассмотрениях и более жесткого управления конфигурацией. На первый взгляд может показаться, что уровень В должен быть существенно дороже уровня С, например на 50-70 %. В теории это выглядит логично, но, как нередко бывает, практика побеждает теорию.

На уровнях В и С должны быть детальные требования низкого уровня к ПО, и они должны быть тщательно протестированы. Напомним, что DO-178C требует детальной верификации требований низкого уровня начиная с уровня С, а эти требования покрывают подавляющее большинство логических решений ПО. При тестировании на основе требований уже покрывается большая часть ветвей исходного кода (80-90 % и более), поэтому дополнительное тестирование структурного покрытия для них не требуется, если во время функционального тестирования правильно используются инструменты записи тестов и покрытия. Поэтому кажущийся значительным рост стоимости из-за структурного покрытия на уровне В по сравнению с уровнем С уже в большой степени компенсируется усиленным тестированием на основе требований по DO-178C. Кроме того, сильные организации разработки ПО уже используют полуавтоматизированный и отлаженный процесс с независимыми рассмотрениями и жестким управлением конфигурацией, поэтому дополнительные затраты на эти аспекты уровня В также в значительной мере сдерживаются.

Те, кто “эффективно” следовал прежнему DO-178B, при переходе на DO-178C увидят еще более заметное влияние на стоимость. Их “эффективность” могла быть связана с вольной трактовкой той детализации требований низкого уровня, которая предполагалась в DO-178B, но контролировалась менее жестко. Такие упрощения позволяли выполнять менее детальное функциональное тестирование с проверкой гораздо меньшего числа логических ветвей. Для DO-178B уровня С это еще было приемлемо, но для DO-178C уровня С уже неприемлемо, потому что DO-178C требует большей детализации требований низкого уровня. Из-за этой большей детализации DO-178C непреднамеренно сократил различие между уровнями С и В: цель структурного покрытия решений и условий для уровня В в значительной степени уже достигается на уровне С за счет более детальных требований низкого уровня. Кроме того, разработчики, широко использующие компоненты параметрических данных (Parameter Data Items) – объекты или логику, внешние по отношению к основной прикладной программе, – теперь обязаны полностью документировать, рассматривать, трассировать и тестировать все эти данные по DO-178C. По замыслу DO-178B им и раньше следовало это делать. Таков итог.

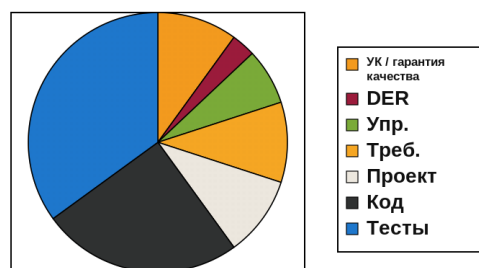
Уровень А – наиболее критичный уровень ПО и потому самый дорогой. Это верно. Но есть еще один миф о логике уровня А: “Уровня А чрезвычайно трудно достичь, и логика будет стоить как минимум на 30-50 % дороже, чем уровень В”. Это неверно. Логика уровня А накладывает дополнительные требования к структурному покрытию, включая модифицированное покрытие условий/решений (Modified Condition/Decision Coverage, MC/DC), корреляцию исходного и двоичного кода, а также большую независимость при рассмотрениях. Самый значимый фактор роста стоимости по сравнению с уровнем В – требование MC/DC. Однако при правильном применении современных инструментов структурного покрытия, обучении персонала и тщательном тестировании на основе требований дополнительные затраты для уровня А можно во многом сдержать, поэтому ПО уровня А лишь немного дороже ПО уровня В. По этой причине большинство готовых коммерческих программных продуктов (Commercial Off-The-Shelf, COTS),

ориентированных на сертифицируемость уровня В, вместо этого выбирают уровень А. Однако, как отмечалось ранее, системные/аппаратурные затраты для уровня А будут выше, чем для уровня В, из-за дополнительного резервирования, необходимого для надежности уровня А, которая в 100 раз выше надежности уровня В (10^{-9} против 10^{-7}).

Куда уходят деньги?

Следующая диаграмма показывает типичную структуру затрат для проекта уровня гарантии разработки (Development Assurance Level, DAL) В; в перечне факторов на рисунке также используется оценка функциональных опасностей (Functional Hazard Assessment, FHA). Разумеется, для уровня А доля верификации была бы больше. Эта диаграмма относится к традиционному структурированному проектированию, а не к разработке на основе моделей (Model-Based Development, MBD). При разработке на основе моделей доля требований и проектирования была бы намного больше, а время кодирования – значительно меньше, если предположить, что применяется автоматическая генерация кода из моделей.

* УК / гарантия качества:	10%
* Услуги DER/CVE:	2-3%
* Управление:	4-7%
* Разработка требований:	10%
* Проектирование:	10%
* Код/логика:	25%
* Верификация:	35%



□ Основные факторы затрат?

1. Назначение DAL / меры снижения по FHA
2. Точные и детальные требования
3. Тщательные рассмотрения для предупреждения дефектов
4. Минимизация изменений кода/логики
5. Эффективное тестирование: меньше повторов, больше автоматизации

Показатели затрат на сертификацию (уровень гарантии разработки В при структурированном проектировании)

Как показано на рисунке выше, пять основных факторов затрат поддерживаются эффективной реализацией DO-XXX и ARP47XX.

Выгоды DO-XXX/ARP47XX.

Как уже показано выше, авиационные руководства не бесплатны и даже не дешевы. Однако при правильной реализации они действительно могут быть экономически эффективными, особенно если оценивать их на протяжении жизненного цикла изделия

или в последующих версиях продукции, когда эффективность и выгоды наиболее заметны. Сводка выгод DO-XXX/ARP47XX приведена на следующем рисунке, а подробности даны далее:

Ключевые выгоды руководств DO-XXX и ARP47XX

1. Раннее предупреждение дефектов и меньше эксплуатационных дефектов.
2. Повышенная возможность повторного использования.
3. Улучшенная видимость и большая уверенность при оценке влияния изменений.
4. Улучшенная видимость для руководства и поставщиков.
5. Количественная и улучшенная оценка безопасности.
6. Снижение затрат, сроков и рисков.
7. Меньше допущений – меньше дефектов.
8. Более полное тестовое покрытие.
9. Более полная документация и более простое сопровождение.
10. Более широкий и глубокий выход на рынок.
11. Повышенное качество.

Эффективные и опытные организации авиационной разработки получают перечисленные выше выгоды DO-XXX/ARP47XX. Ниже они поясняются подробнее.

Большая ясность требований на раннем этапе: DO-XXX/ARP47XX требуют тщательных и детальных требований. Такая детализация и необходимая дисциплина заставляют давать ответы заранее, а не откладывать их на потом. Допущения резко сокращаются. Согласованность требований и их тестопригодность существенно повышаются. Итерации и переделки из-за ошибочных или отсутствующих требований заметно уменьшаются. Да, другие стандарты и руководства, например CMMI, также требуют ранней проработки требований, но DO-XXX отличается именно строгостью обеспечения их детализации.

Меньше итераций реализации: итерации реализации, или постоянные переделки, – бич авиационной инженерии. Например, во многих новых изделиях существует десять, двадцать и даже тридцать версий эволюционирующих файлов кода. Впечатляет, причем не в лучшем смысле. При сильных инженерных процессах и дисциплине реализации должны в значительной мере получаться правильными с первого раза и не требовать десятков обновлений, чтобы “наконец заработало”. Модели и реализацию нужно рассматривать путем анализа их соответствия документированным требованиям.

Меньше ошибок, обнаруживаемых при тестировании разработчиком: поскольку DO-XXX и ARP4754A требуют тщательных и тестопригодных требований, а также рассмотрений кода для ПО уровня C и выше, при тестировании разработчиком

обнаруживается гораздо меньше дефектов. Независимые рассмотрения логики, требуемые для уровней В и А, усиливают этот эффект. Для таких рассмотрений до написания моделей или кода должны быть завершены и сконфигурированы следующие элементы:

- стандарты и контрольные перечни;
- требования высокого и низкого уровня;
- проект ПО;
- трассируемость для всего перечисленного выше.

Большая согласованность внутри ПО: авиационные системы подобны цепи: они не прочнее своего самого слабого звена. Система, корректная на 99 %, некорректна на 1 %, а значит, небезопасна. Системы никогда нельзя доказуемо сделать совершенными, и DO-XXX/ARP47XX не утверждают, что полное выполнение их целей дает совершенные системы. Самый слабый логический модуль или инженер находится на критическом пути безопасности полетов. Все системы должны быть согласованы со своим уровнем критичности, и корректно реализованные DO-XXX/ARP47XX обеспечивают такую согласованность.

Меньше дефектов, обнаруживаемых при интеграции: интеграция может быть долгим итеративным процессом, в ходе которого обнаруживаются и исправляются крупные дефекты, требующие изменений проекта. Но не в DO-178 и DO-278, где интеграция обычно проходит на 50-75 % быстрее, чем в средах без DO-XXX. Поскольку для более высоких уровней гарантии разработки все ПО, работа которого потенциально зависит от аппаратуры, должно тестироваться на реальной целевой аппаратуре, интеграция начинается раньше и ПО верифицируется раньше. Кроме того, система, на которой размещено это ПО, должна соответствовать ARP4754A, требующему столь же сильных системных процессов. Вероятно, пользовательские компоненты аппаратной логики также должны будут соответствовать DO-254, то есть аппаратурному аналогу DO-178C для логики, поэтому и аппаратура к моменту интеграции будет более качественной. В результате интеграция начинается раньше и изначально содержит меньше дефектов, чем в средах без DO-178C.

Улучшенное управление конфигурацией: возможность управлять любым снимком проекта и воспроизводить его обеспечивается управлением конфигурацией. Однако управление конфигурацией также обеспечивает защиту, резервное копирование, завершенность рассмотрений, регистрацию сообщений о проблемах и управление версиями. Строгие требования DO-XXX и ARP47XX к управлению конфигурацией в сочетании с современными инструментами, автоматизирующими многие его задачи, обеспечивают более высокое и более доказуемое качество изделия сейчас и в будущем. Буквально любая версия сертифицированной системы должна полностью воспроизводиться и повторно тестироваться в течение тридцати лет после поставки. Это означает, что под управление и фиксацию должна попадать не только сертифицированная

реализация, но и все вспомогательные файлы: файлы make, сценарии сборки, сторонние инструменты, среды разработки, проектные и тестовые данные и т. д. Такая фиксация устраняет типичные проблемы сопровождения систем, встречающиеся в других отраслях.

Более простое регрессионное тестирование: изделие создается исходя из предположения, что оно будет успешным и поэтому прослужит долго. Вы также знаете, что ПО будет развиваться через новые применения, установки и версии. Регрессионное тестирование может быть дорогим, если оно обширное или выполняется вручную. DO-XXX и ARP4754A дают тщательную трассируемость, позволяющую определить, какие модули требуют изменений или анализа и соответствующего повторного тестирования. Поскольку авиация требует тщательного и доказуемого регрессионного тестирования, сторонники DO-XXX/ARP47XX получают выгоду в виде более простого и обычно автоматизированного повторного тестирования системы, ПО и аппаратуры.

Более тщательное тестирование: 100 %-ное покрытие требований, более ясное и более широкое покрытие робастности, а также структурное покрытие логики являются детерминированными аспектами верификации по DO-XXX. Компоненты параметрических данных должны быть особенно хорошо документированы и верифицированы по DO-178C. Неиспользуемые программные структуры (мертвый код) должны быть удалены, а отключенный код для уровней C, B и A должен быть документирован и верифицирован. Вероятность ошибок ПО, проявляющихся в эксплуатации, резко снижается, особенно на уровнях A и B.

Улучшенное тестирование аппаратуры и системы: системное тестирование обычно сложно для встроенных систем, потому что среда разработки сильно отличается от целевой среды. Кроме того, тестирование часто выполняют разные инженерные группы. Более высокая детерминированность и качество всех этих компонентов благодаря DO-XXX улучшают системную интеграцию. Поскольку DO-XXX и ARP47XX требуют полной трассируемости, через нее вполне можно показать, что системные требования фактически были полностью протестированы во время тестирования ПО и поэтому не требуют повторного дополнительного тестирования на системном уровне.

Меньше дефектов, проявляющихся в эксплуатации: эксплуатационные дефекты и возвраты от заказчиков чрезвычайно дороги и в краткосрочной, и в долгосрочной перспективе. Показано, что изделия, разработанные по DO-XXX, дают на 80-90 % меньше возвратов.

Улучшенное повторное использование: благодаря тщательной и согласованной документации, требуемой DO-XXX, модульности, соблюдению документированных современных инженерных принципов и рассмотрениям, подтверждающим все это, повторное использование существенно улучшается. В авиации повторное использование – это Святой Грааль, но реальность такова: если компонент нельзя повторно использовать хотя бы на 80 % (то есть без изменений), часто быстрее и менее рискованно начать с нуля. На практике большинство ПО пригодно для повторного использования менее чем на 50 %. При DO-XXX, соблюдении стандартов проектирования и кодирования, независимых

рассмотрениях и трассируемости большинство модулей должно быть пригодно для повторного использования как минимум на 70 %. А при применении моделирования и C++ повторное использование возрастает еще сильнее.

Повышенная удовлетворенность заказчиков: надежные системы приводят к надежному возвращению заказчиков. Снижение числа единичных точек отказа – прямой результат DO-XXX. ПО – это искусство; художники сопротивляются документированию своей работы и подчинению ее общим стандартам разработки и коллегиальным рассмотрениям. Без стандартов, дисциплины и современных принципов программной инженерии команды разработки ПО вырождаются в слабо организованную группу самостоятельных художников. Эти художники очень ценны, креативны и талантливы. Но потеря или недостаточная результативность любого такого художника по любой причине становится катастрофой для команды, если его работа не документирована, не понята и не применяется последовательно, как работа остальных. DO-XXX существенно снижает вероятность таких единичных отказов персонала.

Лучшая осведомленность руководства о реальном состоянии графика: сколько авиационных проектов неделя за неделей сообщают статус “готово на 99 %”? Как измеряется ход проекта? Как руководство может действительно определить степень готовности почти невидимой логики? Ответ на все эти вопросы дают современные и достаточно детальные методы управления, построенные вокруг DO-XXX. Авиационные руководства обеспечивают прозрачность, трассируемость и точный статус по проектированию, разработке, тестированию, интеграции и рассмотрениям.

Улучшение сроков завершения за счет всего перечисленного: “что измеряется, то и делается”. DO-XXX/ARP47XX обеспечивают непрерывное отслеживание завершенности проекта и состояния качества.

Более широкое признание на рынке: авиационные руководства дают широко понятное доказательство надежности. Применения, критически важные с точки зрения безопасности, широко распространены в аэрокосмической отрасли, медицине, транспорте, энергетике и даже индустрии развлечений, например в парках аттракционов. Во всех этих областях на кону стоят человеческие жизни, и всем им нужна повышенная, надежно обеспеченная безопасность.

Улучшенная оценка затрат, связанных с изменениями и редакциями: поскольку DO-XXX дает полную видимость внутренней логики, трассируемость требований, проекта, кода и тестов, а также метрики, поддерживаемые гарантией качества, он позволяет яснее определять влияние изменений логики.

Лучшая дифференциация относительно конкурентов: капитализм обеспечивает конкуренцию, а значит, конкуренты будут постоянно перенимать лучшие свойства конкурирующих продуктов, размывая дифференциацию и долю рынка. Что делать? Развивать собственные продукты и самим создавать отличия. Компании и продукты, которые не растут, похожи на деревья, которые не растут: они вскоре погибают.

Авиационные системы резко усложняются, а подготовка разработчиков и инженерная практика едва успевают за этим ростом. Многие компании понимают, что принятие DO-XXX/ARP47XX помогает доказывать надежность, качество и добавленную ценность. Авиационные руководства – это далеко не формальный штамп, они требуют доказательств. “Виновен, пока не доказана невиновность” – таков девиз авиационной надежности и этих руководств.

Повышение качества разделения работ между подрядчиками, субподрядчиками и поставщиками: строгость и полнота процессной документации DO-XXX/ARP47XX помогают избегать недопонимания и ошибок между компаниями, подрядчиками, субподрядчиками и поставщиками, поддерживая эффективное распределение работ между организациями.

Аспекты сертификации беспилотных летательных аппаратов (UAV) и беспилотных авиационных систем (UAS)

Автор: Грег Уилсон, представитель Федерального управления гражданской авиации США (Federal Aviation Administration, FAA) по инженерной экспертизе (Designated Engineering Representative, DER)

Рецензент:

Г-н Вэнс Хилдерман, главный технический директор (Chief Technology Officer, CTO) AFuzion Inc.

За последние 100 лет пилотируемые воздушные суда и их системы прошли большой путь. Автоматизация кабины экипажа существенно выросла, а конструкции авионических систем стали намного сложнее. Такая автоматизация включает электродистанционную систему управления, ЭДСУ (Fly-By-Wire, FBW), автоматическое управление полетом и другие функции автоматизации воздушного судна. Развитие авиационной техники дошло до уровня, при котором вычислители управления полетом часто используются, а иногда и требуются, чтобы дополнять работу экипажа или непосредственно управлять современным воздушным судном.

По мере того как системы пилотируемых воздушных судов становились сложнее и совершеннее, а их роль на борту возрастала, на первый план вышла возможность автономной эксплуатации воздушных судов в смешанном воздушном пространстве. Движение к автономным воздушным судам привело к появлению беспилотных воздушных судов в составе новых конструкций. В течение ряда лет беспилотные авиационные системы, БАС (Unmanned Aircraft Systems, UAS) разрабатывались и эксплуатировались прежде всего военными организациями разных стран. Теперь БАС готовы стать неотъемлемой частью воздушного пространства, которое раньше было предназначено для пилотируемой авиации. В США и во всем мире активно идет работа по включению БАС в воздушное пространство, пока в основном ограниченное эксплуатацией пилотируемых воздушных судов. За последние годы интерес к коммерциализации БАС резко вырос, а стремление коммерческих организаций эксплуатировать их в Национальной системе воздушного пространства США (National Airspace System, NAS) и во многих других странах достигло крайне высокого уровня. Коммерческое применение БАС охватывает аэрофотосъемку, поиск и спасание, разведку, инспекции, например осмотр линий электропередачи, мониторинг мероприятий, репортажную съемку, обнаружение и тушение пожаров, доставку грузов, мониторинг

инфраструктуры, связь и вещание, реагирование на бедствия, сельское хозяйство и многое другое. Это лишь несколько примеров коммерческого применения БАС. По мере интеграции БАС в NAS появятся и новые области применения, о которых пока даже не подумали.

Предпосылки и история

Если учитывать, сколько времени заняло формирование правил и нормативных положений для пилотируемой авиации, то их изменение для допуска БАС к эксплуатации в NAS за сравнительно короткий срок представляет собой серьезную задачу. Разработка правил для пилотируемой авиации всегда была сосредоточена на безопасности полетов. Нормы летной годности для существующих воздушных судов США определены в Своде федеральных нормативных актов США (Code of Federal Regulations, CFR), раздел 14. Процессы FAA по сертификации типа определены в приказе FAA 8110.4, а сертификация летной годности – в приказе FAA 8130.2. Поэтому неудивительно, что изменение этих правил и нормативных положений для включения БАС в NAS было и остается сложным процессом, требующим значительных затрат времени и ресурсов. Сейчас FAA и Агентство Европейского союза по авиационной безопасности (European Union Aviation Safety Agency, EASA) продолжают формировать правила и нормативные положения для эксплуатации БАС, хотя за последние годы в этой работе достигнут заметный прогресс. Организации, которые хотят проектировать, изготавливать, выводить на рынок или эксплуатировать БАС с массой выше минимальных пороговых значений в NAS, должны получить одобрение FAA на БАС: беспилотное воздушное судно, БВС (Unmanned Aerial Vehicle, UAV), пилота БВС, а также эксплуатационную среду и планы эксплуатации БВС. Для одобрения типовой конструкции разработчики БАС должны показать, что конструкция обеспечивает приемлемый уровень безопасности. Эксплуатанты БАС должны использовать сертифицированные FAA системы, позволяющие соблюдать стандартизованные правила воздушного движения, включая процедуры на случай особых и аварийных ситуаций. В рамках работ по БАС FAA согласует подходы с EASA и международным сообществом для завершения формирования правил гражданской авиации и эксплуатации БАС.

Поскольку главный приоритет FAA – безопасность полетов, эксплуатация БАС в NAS в конечном счете потребует выполнения требований FAA по безопасности эксплуатации при полетах в NAS. С начала 1990-х годов БАС допускались к эксплуатации в NAS на ограниченной основе. Сначала такая эксплуатация была ограничена военными и правоохранительными органами. В январе 2013 года FAA выпустило уведомление 8900.207, позднее отмененное и замененное уведомлением N 8900.227 “Unmanned Aircraft Systems (UAS) Operational Approval”. Это уведомление содержало сведения об оценке безопасности БАС, интероперабельности БАС и летной эксплуатации БАС в NAS; его можно использовать при подготовке разрешений (Certificate of Waiver or Authorization, COA), дающих доступ в NAS. Специальный сертификат летной годности относится к категории экспериментальных воздушных судов и требует подтвержденных

возможностей, позволяющих эксплуатировать воздушное судно в NAS. Затем гражданские БАС получили возможность использовать экспериментальные сертификаты по приказу FAA 8130.34. Специальный экспериментальный сертификат летной годности также включает ограничения на эксплуатационное применение воздушного судна. FAA рассматривает каждую заявку БАС на экспериментальный сертификат летной годности отдельно. Такое индивидуальное рассмотрение позволяет FAA аккуратно определить ограниченный уровень доступа, зависящий от мер снижения риска для БАС и ее эксплуатации. Эти ограничения поддерживают безопасную и эффективную эксплуатацию БАС в NAS и не допускают ухудшения доступа других участников к NAS. Применение специального экспериментального сертификата летной годности для БАС аналогично применению экспериментального сертификата для пилотируемых воздушных судов и обычно допускается для научно-исследовательских и опытно-конструкторских работ, НИОКР (Research and Development, R&D), подготовки экипажа или изучения рынка согласно 14 CFR 21.191(a), (c) и (f).

FAA разработало уведомление о предлагаемом нормотворчестве (Notice of Proposed Rulemaking, NPRM), чтобы разрешить малым БАС (small UAS, sUAS) выполнять полеты в NAS. Эта работа включала заинтересованные стороны FAA, связанные с эксплуатацией малых БАС, чтобы выработать консенсусные стандарты, необходимые для нормотворчества и применения новых правил. Затем FAA работало с Международной организацией гражданской авиации, ИКАО (International Civil Aviation Organization, ICAO), специализированным учреждением ООН, которое содействует безопасному и упорядоченному развитию международной гражданской авиации во всем мире. ИКАО устанавливает стандарты и нормативные положения в области безопасности полетов и авиационной безопасности. ИКАО рассмотрела БАС, чтобы создать базовую международную нормативную основу для поддержки их эксплуатации во всем мире. ИКАО выпустила руководящий материал, требующий от государств-членов внедрения программ системы управления безопасностью полетов, СУБП (Safety Management System, SMS). Эти программы необходимы для управления рисками, связанными с применением БАС в авиационной системе. Эксплуатация БАС создает много новых задач, включая анализ огромных объемов данных для получения полезной информации, необходимой для надзора и оценки риска. Циркуляр ИКАО 328 содержал руководящий материал по введению и интеграции БАС в воздушное пространство. Цель, определенная в этом циркуляре, состоит в обеспечении интероперабельности и нормативной совместимости там, где это возможно. Затем FAA работало с комитетами, например с Радиотехнической комиссией по аэронавтике (Radio Technical Commission for Aeronautics, RTCA), специальными комитетами SC-203 и SC-228, чтобы определить минимальные стандарты характеристик (Minimum Performance Standards, MPS) и минимальные эксплуатационные стандарты характеристик (Minimum Operational Performance Standards, MOPS) для эксплуатации БАС в NAS США. После этого FAA опубликовало документ с дорожной картой и планами интеграции БАС в NAS. Дорожная карта FAA по БАС называлась “Integration of Civil Unmanned Aircraft Systems (UAS) in the National Airspace System (NAS) Roadmap”, первое издание 2013 года, опубликована в ноябре 2013 года. Эта дорожная

карта была разработана FAA и Комитетом по нормотворчеству в области авиации (Aviation Rulemaking Committee, ARC), чтобы изложить планы и цели FAA по интеграции БАС в NAS. В дорожной карте были определены общие сроки, задачи и соображения, необходимые для такой интеграции. Затем комитеты продолжили разработку минимальных авиационных системных стандартов характеристик (Minimum Aviation System Performance Standards, MASPS) для БАС и разработку MOPS, используемых в поддержку технических стандартных приказов (Technical Standard Orders, TSO) в среднесрочной и долгосрочной перспективе.

Как и при разработке пилотируемых воздушных судов и их систем, большинство БАС проходят процесс, во многом похожий на процесс разработки пилотируемой авиационной техники. Одно из отличий связано с оценкой обоснования безопасности БАС. Прежде чем приступить к разработке требований к БАС, необходимо подготовить полное обоснование безопасности эксплуатации БАС. Затем это обоснование становится основой для определения оценки безопасности системы, включая уровни гарантии разработки ПО по DO-178C и уровни гарантии разработки аппаратуры по DO-254 для бортовой электронной аппаратуры. Как и в случае пилотируемых воздушных судов и систем, оценка безопасности системы БАС будет опираться на SAE ARP4754A “Guidelines for Development of Civil Aircraft and Systems” и SAE ARP4761 “Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment”. Дополнительный руководящий материал, полезный при подготовке обоснования безопасности, содержится в стандартизационном соглашении НАТО (Standardization Agreement, STANAG) 4671 “Unmanned Aerial Vehicle Systems Airworthiness Requirements (USAR)” и STANAG 4586 “Standard Interfaces of UAV Control System (UCS) for NATO UAV Interoperability”. Если цели, применяемые к пилотируемым воздушным судам при подготовке обоснования безопасности и оценки безопасности, напрямую перенести на БАС, можно получить систему с уровнем безопасности выше фактически необходимого. С другой стороны, если цели безопасности для БАС задать слишком низко, разработка может не обеспечить достаточный уровень безопасности и, следовательно, может не выполнить требования и нормативные положения FAA к БАС, включая MPS/MOPS.

Наряду с обычными сложностями разработки любой системы обоснование безопасности БАС должно охватывать следующее.

Пилот и экипаж БАС:

- подготовка;
- эксплуатационные стандарты;
- сертификационные требования;
- эксплуатационные процедуры;
- нормативные требования;
- медицинские стандарты;
- стандарты испытаний.

Станция управления:

- сертификационные требования и базис сертификации, например DO-178C и/или DO-278A “Guidelines for Communication, Navigation, Surveillance, and Air Traffic Management (CNS/ATM) Systems Software Integrity Assurance”;
- технические стандартные приказы (Technical Standard Orders, TSO), где применимо;
- нормы летной годности;
- нормативные положения;
- требования к интероперабельности;
- руководящие материалы, включая консультативные циркуляры, приказы, уведомления и т. д.;
- поддержание летной годности;
- методы подтверждения соответствия (Means of Compliance, MOC);
- голосовое командно-диспетчерское взаимодействие с органами управления воздушным движением FAA, УВД (Air Traffic Control, ATC).

Канал передачи данных:

- сертификационные требования и базис сертификации, например DO-178C и/или DO-278A “Guidelines for Communication, Navigation, Surveillance, and Air Traffic Management (CNS/ATM) Systems Software Integrity Assurance”;
- технические стандартные приказы, где применимо;
- нормы летной годности;
- требования к интероперабельности, где применимо;
- руководящие материалы, включая консультативные циркуляры, приказы, уведомления и т. д.;
- координация авиационного радиочастотного спектра и применение стандартизованных архитектур управления;
- показатели характеристик;
- требования к защите радиоканала и канала передачи данных;
- методы подтверждения соответствия;
- нормативные положения.

Беспилотное воздушное судно:

- сертификационные требования и базис сертификации;
- технические стандартные приказы;
- нормы летной годности;
- процедуры;
- нормативные положения;
- руководящие материалы, включая консультативные циркуляры, приказы, уведомления и т. д.;

- показатели характеристик;
- поддержание летной годности;
- стандарты и записи испытаний;
- методы подтверждения соответствия.

Дополнительная информация и руководящий материал доступны в DO-304 “Guidance Material and Considerations for Unmanned Aircraft Systems”, DO-320 “Operational Services and Environmental Definition (OSED) for Unmanned Aircraft System” и DO-344, тома 1 и 2, “Operational and Functional Requirements and Safety Objectives for Unmanned Aircraft Systems Standards”. Нормативные положения FAA основаны на правилах для пилотируемых воздушных судов. Исторически эти правила требовали, чтобы командир воздушного судна, КВС (Pilot In Command, PIC) следовал принципу “see and avoid” – “видеть и избегать” – и постоянно поддерживал ситуационную осведомленность. Появление БАС требует перехода от “see and avoid” к “sense and avoid” – “обнаруживать и избегать”. В пилотируемой авиации принцип “видеть и избегать” вместе с радиолокацией, визуальным наблюдением, стандартами эшелонирования и проверенными технологиями обеспечивал безопасную эксплуатацию пилотируемых воздушных судов. Для БАС требуется смена парадигмы: переход от “видеть и избегать” к “обнаруживать и избегать”. Эта смена делает особенно важным комплекс датчиков БАС, наземных и бортовых, поскольку именно он должен заменить функцию КВС пилотируемого воздушного судна по поддержанию ситуационной осведомленности. Кроме того, роль линии управления и связи C2 (Control and Communication, C2) также требует переосмысления, поскольку БАС зависят от наземного канала C2.

Линия управления и связи C2 для БАС также развивается: ведутся значительные НИОКР, направленные на оценку жизнеспособности C2 для БАС. Канал C2 между беспилотным воздушным судном и наземной станцией управления требует определения требований к C2 и требуемых характеристик C2. Сейчас ведется большая работа по оценке каналов передачи данных управления БАС, включая задержку и связанные с ней проблемы, целостность, непрерывность, готовность, подавление помехами, функциональные отказы и/или резервные системы, а также многие другие критерии характеристик. Результатом этой работы являются:

- определение требований к C2 для БАС;
- определение критериев характеристик C2 для БАС;
- оценка и определение уязвимости и защиты канала C2 для БАС;
- оценка и определение спектра связи C2 для БАС, что требует координации с американскими и международными организациями радиосвязи;
- определение мер снижения риска;
- установление критериев сквозного измерения характеристик БАС;
- разработка концепций следующего поколения для критериев четырехмерной траектории БАС.

В поддержку работ по C2 для БАС несколько проектов НИОКР сосредоточены на разработке и валидации систем каналов управления БАС, количественной оценке критически важных с точки зрения безопасности систем связи БАС и моделировании характеристик C2. Разработка C2 для БАС является совместной работой FAA, государственных ведомств США, промышленности и международных заинтересованных сторон.

Система обнаружения и уклонения (Sense and Avoid, SAA) для БАС также развивается: ведутся значительные НИОКР, чтобы оценить жизнеспособность SAA для БАС. В рамках этих работ оцениваются комплексы датчиков и типы датчиков, как бортовые, так и наземные. Основная часть работ по SAA для БАС сосредоточена на оценке применения нескольких датчиков и интеграции разных датчиков в составной комплекс, способный удовлетворить требования SAA для БАС. В основном SAA относится к одной из двух категорий: наземная SAA и бортовая SAA, то есть комплекс датчиков БВС. Разработка SAA для БАС – очень сложная задача с уникальными и непростыми проблемами. Дополнительную сложность создает нехватка требований. Чтобы закрыть этот пробел, FAA провело исследования SAA для БАС. Эти исследования включали следующие задачи:

- определить и зафиксировать определения системы SAA для БАС и цели по ее характеристикам;
- оценить разные реализации SAA для БАС, многодатчиковые системы SAA и применение нескольких технологий;
- определить и установить минимальный набор информации, необходимый для маневрирования БАС с целью предотвращения столкновений.

Исследования продолжаются, включая фактические испытательные полеты наземной и бортовой SAA для БАС. Из-за сложности бортовой SAA ожидается, что эта технология полностью сформируется лишь через несколько лет, поэтому наземная SAA может стать первой практически применяемой формой SAA для БАС.

Еще одна тема, которую следует обсудить, – человеческие факторы (Human Factors, HF) применительно к БАС. Смена парадигмы, возникающая при входе БАС в воздушное пространство, ранее отведенное пилотируемым воздушным судам, приведет к новым сложным вопросам, связанным с обменом данными между наземным пилотом БАС, органами УВД в NAS и наземным управлением. Для этого потребовались сбор и анализ взаимодействия пилотов БАС с УВД при эксплуатации БАС в воздушном пространстве, где работают и другие пользователи. Это очевидно сложная тема: сбор и анализ такого взаимодействия с УВД в NAS, как можно ожидать, требуют фактической эксплуатации БАС в NAS наряду с другими пользователями гражданского воздушного пространства. Объединенное управление США по планированию и развитию (Joint Planning and

Development Office, JPDO) совместно с академическими организациями, промышленностью и исследовательскими центрами выявило несколько проблем, связанных с человеческими факторами БАС, включая следующие:

- человеческие факторы, относящиеся к требованиям к характеристикам БАС, УВД и т. д.;
- человеческие факторы, относящиеся к квалификации и критериям или требованиям для получения статуса пилота БАС;
- человеческие факторы, относящиеся к выявлению и определению обмена информацией между пилотом БАС, наземным диспетчером БАС, УВД, диспетчером полетов и т. д.;
- человеческие факторы, относящиеся к эксплуатации БАС при потере канала управления, потере связи и рабочих нагрузках пилота БАС, наземного диспетчера и УВД;
- человеческие факторы, относящиеся к эшелонированию воздушных судов органами УВД и отображению этой информации;
- человеческие факторы, относящиеся к наземной станции управления (Ground Control Station, GCS), ее индикаторам, применению и эксплуатации;
- человеческие факторы, относящиеся к ситуационной осведомленности БАС.

Как уже отмечалось, оценка и анализ человеческих факторов для БАС требуют сбора и оценки данных о фактических полетах БАС в NAS наряду с другими пользователями гражданского воздушного пространства. Поскольку эксплуатация БАС в NAS опирается на анализ взаимодействия пилотов БАС с УВД, потребность в фактических данных о таком взаимодействии могут закрыть некоторые ограниченные полеты военных БАС в ограниченном воздушном пространстве, а также другие данные, которые можно получить с использованием запланированных испытательных полигонов FAA для БАС. Минимальный набор данных, который потребуется собрать, включает:

- сбор данных об операциях пилотов БАС;
- анализ того, как УВД будет предоставлять обслуживание пользователям воздушного пространства, включая пилотируемые и беспилотные воздушные суда;
- анализ взаимодействия диспетчеров УВД с пилотом БАС;
- анализ автоматизации и способов ее применения.

Наряду с пилотом БАС и системами БАС FAA также разработало изменения для процессов организации воздушного движения (Air Traffic Operations, АТО). Эксплуатация БАС создает уникальные и сложные условия, которые ранее не требовалось учитывать при эксплуатации пилотируемых воздушных судов. Поэтому многие политики и процедуры воздушного движения потребуется пересмотреть, переписать или, в некоторых случаях, создать с нуля. Изменение этих процессов и процедур воздушного движения

будет частично опираться на исследования, анализ и моделирование эксплуатации БАС в NAS. Ниже перечислены ключевые процессы и процедуры организации воздушного движения, изменение которых нужно было рассмотреть.

Диспетчер:

- руководства по политике;
- подготовка;
- маршрутные, терминальные и океанические операции.

Эксплуатация:

- политика;
- управление воздушным движением;
- планирование полетов;
- эшелонирование и управление потоками;
- нормальные процедуры;
- процедуры на случай непредвиденных обстоятельств;
- возврат на базу;
- потеря управления;
- потеря связи / потеря телеметрии;
- маршрутные, терминальные и океанические процедуры;
- интеграция БАС в наземные операции аэропорта.

Безопасность:

- сбор и анализ данных;
- обоснование безопасности;
- требования безопасности;
- оценка после внедрения;
- маршрутные, терминальные и океанические процедуры.

Интеграцию БАС в NAS дополнительно усложняет широкий диапазон БАС, которые, как ожидается, будут запрашивать доступ в NAS. Конструкции БАС варьируются от очень малых до очень больших и имеют сильно различающиеся характеристики и требования. Сейчас нормативные требования к голосовым ответам пилотируемых воздушных судов на указания УВД не задают количественное время ответа. Поскольку действующие правила не включают время ответа при голосовой связи, для БАС это потенциально потребовало включить время отклика связи непосредственно в критерии проектирования системы БАС. Эксплуатация БАС в NAS добавляет определенную эксплуатационную задержку между БАС и обменом голосовой связью между оператором БАС и УВД. Кроме того, требования к линии управления и связи C2, которые сейчас определяются, будут включать использование сторонних служб связи, уже являющихся неотъемлемой частью эксплуатации пилотируемых воздушных судов.

В заключение: интеграция БАС в NAS была и остается чрезвычайно сложной задачей. До допуска БАС к полетам в NAS нужно было решить множество вопросов, и эта работа продолжается. Краткосрочная, среднесрочная и долгосрочная работа FAA по эксплуатации БАС в NAS сосредоточена на безопасности полетов. Краткосрочный доступ БАС в NAS основан на улучшенных процедурах и технологических достижениях, которые ускорили предоставление такого доступа. В краткосрочной перспективе НИОКР в области передовых мер снижения риска помогли FAA обеспечить включение БАС в NAS. Эти НИОКР вместе с соответствующими ограничениями и условиями предназначены для снижения последствий любых недостатков характеристик БАС и поддержки краткосрочной интеграции БАС в NAS.

Заключение

“Это была темная и бурная ночь... которая в конце концов сменилась солнечным светом.”

Все когда-нибудь заканчивается, пусть иногда этот конец и оказывается временным. Лес, как постоянно развивающаяся экосистема, никогда не бывает окончательно завершен. Мудрый лесничий лишь достаточно хорошо понимает происходящее, чтобы в общих чертах предвидеть будущие изменения и направлять их. Точно так же ни одна книга о сложных авиационных системах не может быть по-настоящему завершенной: это противоречило бы динамичной природе технологической экосистемы. Зато теперь у вас есть базовые инструменты, чтобы понять этот непрерывный процесс и помочь направить будущее авиации к лучшему урожаю.

Некоторые читатели, несомненно, хотели бы получить книгу, которая ответит на все их вопросы. Конечно. Если не считать того, что число возможных вопросов бесконечно, а объем этой книги ограничен, такой безудержный оптимизм и впрямь достоин аплодисментов. Но вернемся к реальности: еще со времен Сократа хорошие учителя знали, что они наиболее полезны тогда, когда закладывают основу, направляют и поощряют подлинное обучение, которое лишь косвенно связано с обычным преподаванием. Если нужно быстро утолить физический голод, достаточно воспользоваться результатом работы повара. Если же нужно защититься от голода на всю жизнь, задача иная: придется освоить основы приготовления пищи, и тогда при наличии навыков и ресурсов вы будете сыты всегда. Поэтому эта книга дает базовую основу знаний об авиационной разработке и может служить ориентиром в поиске дальнейших ответов, чтобы вы продуктивнее участвовали в собственном техническом росте. Возможно, “великая” книга на тысячи страниц могла бы ответить на многие вопросы читателя. Хорошая книга дает читателю средства, чтобы он мог отвечать на такие вопросы сам. Надеюсь, это хорошая книга, потому что я уверен: она не настолько “великая”. Инженеры просто не находят времени читать тысячи страниц.

Мир высоконадежных и критически важных с точки зрения безопасности систем огромен и растет экспоненциально. Об этой теме нетрудно было бы выпускать том за томом, даже если ограничиться базовыми вопросами разработки и сертификации авионики. Вскоре результат стал бы похож на энциклопедическое собрание, которое прекрасно собирает пыль на полке, пока нечитающая аудитория заново изобретает изложенные в нем истины. Говорят, что блестящий Марк Твен однажды написал необычно длинное личное письмо и начал его с извинения: “Прошу прощения, у меня не

было времени сделать его короче”. То же верно и для этой книги, хотя были приложены огромные усилия, чтобы сделать собранный здесь материал кратким, уместным, полезным и, главное, коротким.

Надеюсь, читать эту книгу вам действительно было интересно и полезно. Помните: мир тесен, и тем больше причин жить широко.

Безопасного неба,

Вэнс Хилдерман

Высказывания Вэнса Хилдермана, начиная с 1989 года, когда он основал свою первую компанию, и до наших дней, уже после седьмой. Они приведены по электронным письмам и интервью с несколькими десятками из более чем 500 его бывших сотрудников.

- *“Система может быть безопасной лишь настолько, насколько вы можете это доказать... но обычно даже меньше.”*
- *“Если это можно придумать, спроектировать и реализовать меньше чем за месяц, вероятно, это не сложная система.”*
- *“ПО не зря по-английски называется software: без должного ограничения сожмете его – и оно окажется на вас.”*
- *“Аппаратура не зря по-английски называется hardware: если она сломалась, чинить ее трудно.”*
- *“Качество – это образ мышления, а не дополнительный компонент.”*
- *“ПО может напоминать зоопарк: хорошим программным зверям нужны месяцы, чтобы создать вроде бы приемлемую систему, а обезьяна может обрушить ее за мгновение.”*
- *“Между простым и сложным есть простое различие: простое требует компетентности, а сложное требует дисциплинированной компетентности.”*
- *“Компании, которым выручка интереснее долгосрочного успеха бизнеса, в конце концов не получают ни того, ни другого.”*
- *“Положительные действия МОГУТ дать положительные результаты; отрицательные действия ОБЯЗАТЕЛЬНО дадут отрицательные результаты.”*
- *“Нанимая инженеров, учитывайте и опыт, и прежние ошибки: первое помогает свести второе к минимуму.”*
- *“Учиться чинить автомобильные тормоза очень увлекательно, поэтому сначала стоит попробовать это на машине друга.”*
- *“Дела говорят громче слов, а факты говорят громче дел.”*

- *“Базовая математика – необходимый навык для успеха: любой предприниматель легко посчитает, что половина рабочего дня равна двенадцати часам.”*
- *“Всегда просите соискателя на авиационную должность принести одностраничное резюме на бумаге: важнейшее его применение – проверить, как быстро кандидат сделает из него бумажный самолетик.”*
- *“Когда ученик укротителя львов говорит мастеру: ‘Я не боюсь’, но приносит ружье и отказывается идти первым, смысл только один: он осторожно боится.”*
- *“Нанимайте кандидата на миллион долларов: того, чья последняя ошибка стоила прежнему работодателю этой суммы, потому что такая ошибка почти наверняка останется единственной.”*
- *“Когда большое правительство закрывается на день, а фондовые рынки в этот день растут, истинная ценность большого правительства становится очевидной.”*
- *“Прощайте прошлое, но не забывайте его: чтобы жить завтрашним днем, полезно помнить уроки вчерашнего.”*
- *“Если одна картинка стоит тысячи слов, то хорошее объяснение стоит тысячи картинок.”*
- *“Сосредоточьтесь на моменте: когда пьете вино, не вспоминайте пиво...”*
- *“Мир тесен; тем больше причин жить широко.”*

Приложение А: Сокращения в авионике

У авионики есть собственный язык. Много лет назад моя жена Коллин сопровождала меня на авиационный симпозиум, где я выступал, и позже сказала: “Ваши авиационные инженеры говорят не по-английски... во всяком случае, не на том английском, который понимают обычные люди”. Так и есть. Когда слушаешь разговор инженеров по авионике или читаешь литературу по разработке авиационных систем, легко запутаться в этом языке – “авианглийском”, или “авиационном английском”. Но, как и в любом языке, базовый словарь и немного контекста быстро снимают ореол таинственности. В предыдущих книгах и статьях я так и не собрал вместе множество сокращений, которыми полны такие тексты; многие читатели просили подготовить простой список. Он всегда был в моем списке дел, сразу после чистки двигателя моей машины. Итак, ниже приведены наиболее распространенные сокращения в области авиационной разработки; а теперь – чистить двигатель...

Безопасного неба,

Вэнс Хилдерман

Вэнс Хилдерман vance.hilderman@afuzion.com

Сокращение	Определение
A/D	Аналого-цифровой
A/P	Автопилот
ABAS	Бортовая система функционального дополнения
ABROAD	Трансляция данных автоматического зависимого наблюдения
AC	Консультативный циркуляр
ACAS	Бортовая система предупреждения столкновений
ACARS	Система адресации и передачи сообщений воздушного судна
ACC	Районный диспетчерский центр
ACE	Электроника управления приводом
ACMS	Система мониторинга состояния воздушного судна
ACO	Офис сертификации воздушных судов (обычно Федерального управления гражданской авиации США)
ACP	Панель управления аудиосистемой
ACS	Система управления аудиосистемой
AD	Директива летной годности
ADAHRS	Система воздушных данных, пространственного положения и курса
ADC	Вычислитель воздушных данных или аналого-цифровой преобразователь
ADF	Автоматический радиокompас
ADI	Командно-пилотажный индикатор пространственного положения
ADIRS	Инерциальная система воздушных данных
ADIRU	Блок инерциальной системы воздушных данных
ADM	Модуль воздушных данных
ADR	Прием бортовых данных
ADS	Автоматическое зависимое наблюдение
ADS-A	Автоматическое зависимое наблюдение-адресное
ADS-B	Автоматическое зависимое наблюдение-вещание, АЗН-В
ADS-C	Автоматическое зависимое наблюдение-контрактное
AEH	Бортовая электронная аппаратура
AESA	Активная электронно-сканируемая решетка
AFCS	Система автоматического управления полетом
AFD	Автопилот и командный пилотажный прибор
AFDC	Вычислитель автопилота и командного пилотажного прибора
AFDS	Система автопилота и командного пилотажного прибора
AFDX	Полнодуплексная коммутируемая сеть авионики
AGACS	Автоматический наземный

Сокращение	Определение
AGDL	Канал передачи данных “воздух-земля”
AHC	Управление пространственным положением и курсом
AHRS	Система определения пространственного положения и курса
AIDS	Интегрированные данные воздушного судна
AIM	Руководство по аэронавигационной информации
AIP	Сборник аэронавигационной информации
AIR	Информационный отчет по аэрокосмической отрасли
AIS	Информационная система для пилотов
AL	Уровень гарантии
ALT	Высота
AMC	Приемлемый метод определения соответствия (иногда: альтернативный метод определения соответствия)
AMS	Система управления воздушной средой
ANC	Активное шумоподавление
ANN	Сигнализатор
ANR	Активное снижение шума
ANSI	Американский национальный институт стандартов
ANT	Антенна
AOC	Эксплуатационное управление воздушным сообщением
AOP	План эксплуатации аэропорта
AOPA	Ассоциация владельцев воздушных судов и пилотов
APARS	Система автоматической передачи данных о барометрической высоте
APS	Система автопилота
APU	Вспомогательная силовая установка
APV	Заход на посадку с вертикальным наведением
ARINC	Корпорация авиационной радиосвязи
ARP	Рекомендуемая аэрокосмическая практика (публикация SAE)
ASA	Оценка безопасности воздушного судна
ASD	Индикатор воздушной обстановки
ASDL	Спутниковый канал передачи аэронавигационных данных
ASIC	Специализированная интегральная схема
ASR	Аэродромный обзорный радиолокатор
ASTERIX	Универсальный структурированный обмен информацией наблюдения Европейской организации по безопасности аэронавигации
ASU	Блок коммутации авионики

Сокращение	Определение
АТ	Автотяга
АТС	Измененный сертификат типа
АТС	Управление воздушным движением
АТСRBS	Радиомаячная система УВД
АТСSS	Сигнальная система управления воздушным движением
АТСТ	Аэродромно-диспетчерская вышка
АТIS	Автоматическая служба терминальной информации
АТМ	Организация воздушного движения
АТС	Служба воздушного движения
АТСАW	Ситуационная осведомленность о воздушном движении на борту
АТСU	Блок служб воздушного движения
АТТ	Пространственное положение
Avionics	Авиационная электроника
AVN	Управление стандартов авиационных систем
AWACS	Авиационная система дальнего радиолокационного обнаружения и управления
AWOS	Автоматизированная система метеонаблюдения
B RNAV	Базовая зональная навигация
BARO	Барометрическая индикация, установка или давление
BDI	Индикатор пеленга и дальности
BGAN	Глобальная широковещательная сеть покрытия
BIT	Встроенный контроль
BITE	Средства встроенного контроля
BOM	Ведомость материалов
CAA	Управление гражданской авиации
CAS	Калиброванная воздушная скорость
CAST	Группа сертификационных органов по ПО
CAT	Катастрофический (в контексте безопасности полетов; в других случаях может означать “категория”)
CAT I	Эксплуатационная категория характеристик I
CAT II	Эксплуатационная категория характеристик II
CAT IIIa	Эксплуатационная категория характеристик IIIa
CAT IIIb	Эксплуатационная категория характеристик IIIb
CAT IIIc	Эксплуатационная категория характеристик IIIc
CC	Категория контроля
CC	Управление изменениями

Сокращение	Определение
CCA	Анализ общих причин или печатная плата в сборе
CDA	Заход на посадку по непрерывному снижению
CDI	Индикатор отклонения от курса
CDR	Критический анализ проекта
CDRL	Перечень требований к контрактным данным
CDTI	Кабинный индикатор информации о воздушном движении
CDU	Блок управления и индикации
CEH	Сложная электронная аппаратура
CFIT	Управляемый полет со столкновением с землей
CFR	Свод федеральных нормативных актов США
CI	Единица конфигурации
CIDS	Система данных внутрисалонной связи
CM	Управление конфигурацией
CMA	Анализ общего режима отказов
CMP	План управления конфигурацией
CMR	Требование по поддержанию сертификации
CNS	Связь, навигация, наблюдение
CODEC	Кодер/декодер
COMM	Приемник связи
COTS	Готовое коммерческое изделие
CPDLC	Линия передачи данных “диспетчер – пилот”
CPS	Циклов в секунду
CPU	Центральный процессор
CRC	Циклическая проверка избыточности или циклический избыточный код
CRT	Электронно-лучевая трубка
CS	Сертификационная спецификация
CSCI	Единица конфигурации компьютерного ПО
CTAF	Общая консультативная частота воздушного движения
CV/DFDR	Бортовой речевой самописец и цифровой регистратор полетных данных
CVE	Инженер по верификации соответствия
CVR	Бортовой речевой самописец
CWS	Управление от штурвала
DA	Высота принятия решения
DA	Угол сноса

Сокращение	Определение
DAL	Уровень гарантии разработки (или проектирования)
DAPs	Передача параметров воздушного судна по нисходящему каналу
DAR	Назначенный представитель по летной годности
DC	Постоянный ток или покрытие решений
DCDU	Блок управления и индикации канала передачи данных
DCN	Извещение об изменении документа
DCP	Панель управления индикацией
DD	Диаграмма зависимостей
DDS	Док-станция дисплея
DER	Представитель FAA по инженерной экспертизе
DFMC	Двухчастотная многосозвездная система
DG	Курсовой гироскоп
DGPS	Дифференциальная глобальная система позиционирования
DH	Высота принятия решения
DL	Канал передачи данных
DLR	Регистратор канала передачи данных
DME	Дальномерное оборудование
DMIR	Назначенный представитель по производственному инспектированию
DNC	Прямое шумоподавление
DO	Документ RTCA (“Document Order”, порядковый номер документа)
DOD	Министерство обороны
DP	Процедуры вылета
DSP	Процессор цифровой обработки сигналов
DUAT	Терминал прямого пользовательского доступа
DVE	Среда с ухудшенной визуальной видимостью
EADI	Электронный командно-пилотажный индикатор пространственного положения
EASA	Агентство Европейского союза по авиационной безопасности
ECN	Извещение об инженерном изменении
ECU	Блок управления двигателем (= электронное управление двигателем)
EEC	Электронное управление двигателем
EEPROM	Электрически стираемое программируемое постоянное запоминающее устройство
EFB	Электронный полетный планшет
EFD	Электронный пилотажный индикатор
EFIS	Электронная система индикации полетной информации

Сокращение	Определение
EGPWS	Усовершенствованная система предупреждения опасного сближения с землей
EGT	Температура выхлопных газов
EHS	Расширенное наблюдение
EHSI	Электронный индикатор горизонтальной обстановки
EIA	Ассоциация электронной промышленности
EICAS	Система индикации параметров двигателя и предупреждения экипажа
ELT	Аварийный радиомаяк-локатор
EMI	Электромагнитные помехи
EMS	Служба экстренной медицинской помощи
ENC	Электронное шумоподавление
ENG	Двигатель
ENR	Электронное снижение шума
EOC	Исполняемый объектный код
EPR	Степень повышения давления в двигателе
EPROM	Стираемое программируемое постоянное запоминающее устройство
ETOP(S)	Эксплуатация двухдвигательных воздушных судов увеличенной дальности
ETSO	Европейский технический стандартный приказ
EUROCAE	Европейская организация по оборудованию для гражданской авиации
FAA	Федеральное управление гражданской авиации США
FADEC	Полновластная цифровая система управления двигателем
FAI	Инспекция первого изделия
FANS	Перспективная аэронавигационная система
FAR	Федеральные авиационные правила
FBW	Электродистанционная система управления, ЭДСУ
FC	Отказное состояние
FCC	Вычислитель управления полетом
FCR	Итоговое рассмотрение сертификации
FCS	Система управления полетом
FD	Командный пилотажный прибор
FDAL	Уровень гарантии разработки функции
FDE	Обнаружение и исключение отказов
FDPS	Система обработки данных плана полета
FDR	Регистратор полетных данных
FDRS	Система регистрации полетных данных

Сокращение	Определение
FDU	Блок датчика магнитного потока
FF	Расход топлива
FFR	Рассмотрение первого полета
FFS	Набор функциональных отказов
FG	Управление полетом
FHA	Оценка функциональных опасностей
FIC	Центр полетной информации
FIFO	“Первым пришел – первым вышел”
FIS	Служба полетной информации
FIS-B	Службы полетной информации
FL	Эшелон полета
FLIR	Инфракрасная система переднего обзора
FLTA	Предупреждение о рельефе в направлении полета
FM	Формальные методы
FMA	Сигнализатор режима полета
FMEA	Анализ видов и последствий отказов
FMES	Сводка видов и последствий отказов
FMGS	Система управления полетом и наведения
FMS	Система управления полетом
FOB	Топливо на борту
FPE	Эмуляция плавающей точки
FPGA	ПЛИС, программируемая логическая интегральная схема
FREQ	Частота
FSS	Станция полетного обслуживания
FTA	Анализ дерева отказов
FWS	Система предупреждения экипажа
FYDS	Система командного пилотажного прибора и демпфера рыскания
G/S	Глиссада
GA	Авиация общего назначения
GA	Уход на второй круг
GAST	Тип службы захода с использованием наземной системы функционального дополнения
GBAS	Наземная система функционального дополнения
GCAS	Система предупреждения столкновения с землей
GCU	Блок управления генератором

Сокращение	Определение
GDOP	Геометрический фактор ухудшения точности
GGS	Наземная станция глобальной системы позиционирования
GHz	Гигагерц
GLNS	Система посадки и навигации глобальной системы позиционирования
GLNU	Блок системы посадки и навигации глобальной системы позиционирования
GLONASS	Глобальная навигационная спутниковая система
GLS	Система посадки с использованием наземной системы функционального дополнения
GLS	Система посадки с использованием глобальной навигационной спутниковой системы
GLU	Посадочный блок глобальной системы позиционирования
GMT	Среднее время по Гринвичу
GND	Земля
GNSS	Глобальная навигационная спутниковая система, ГНСС
GO	Уход на второй круг
GPIO	Универсальный ввод/вывод
GPS	Глобальная система позиционирования
GPWC	Вычислитель системы предупреждения опасного сближения с землей
GPWS	Система предупреждения опасного сближения с землей
GS	Путевая скорость или наземная станция
HAS	Сводка по аппаратуре
HAZ	Опасность или опасный
HCI	Указатель конфигурации аппаратуры
HCM	Управление конфигурацией аппаратуры
HCMP	План управления конфигурацией аппаратуры
HDG	Курс
HDG SEL	Выбор курса
HDL	Язык описания аппаратуры
HDOP	Горизонтальный фактор ухудшения точности
HDP	План разработки аппаратуры
HF	Высокая частота
HHL D	Удержание курса
HIRF	Поле высокой интенсивности излучения
HMD	Нашлемный индикатор
HMI	Человеко-машинный интерфейс
HOL	Язык высокого уровня

Сокращение	Определение
HPAP	План процесса гарантии аппаратуры
HPR	Ротор высокого давления
HSD	Высокоскоростные данные
HSI	Индикатор горизонтальной обстановки
HSL	Выбор курса
HTAWS	Вертолетная система предупреждения о рельефе и опасных режимах
HUD	Индикатор на лобовом стекле
HUMS	Системы мониторинга технического состояния и эксплуатации
HVVP	План верификации и валидации аппаратуры
HW или H/W	Аппаратура
Hz	Герц
I/O	Ввод/вывод
IAS	Приборная скорость
ICA	Инструкции по поддержанию летной годности
ICAO	Международная организация гражданской авиации
ICD	Документ контроля интерфейса
ICE	Внутрисхемный эмулятор
ID	Идентифицировать/идентификация или идентификатор
IDAL	Уровень гарантии разработки элемента
IDE	Интегрированная среда разработки
IDENT	Идентифицировать/идентификатор
IDS	Система отображения информации или интегрированная система отображения
IFE	Бортовая развлекательная система
IFF	Свой-чужой
IFICS	Интегрированная система пилотажных приборов и управления
IFR	Правила полетов по приборам
ILS	Инструментальная система посадки
IMA	Интегрированная модульная авионика
IMC	Метеоусловия полета по приборам
IND	Индикатор
INS	Инерциальная навигационная система
IOC	Начальная эксплуатационная готовность
IP	Интернет-протокол или интеллектуальная собственность
IPV	Схема захода по приборам с вертикальным наведением (позднее переименована в заход на посадку с вертикальным наведением)

Сокращение	Определение
IRS	Инерциальная опорная система или спецификация требований к интерфейсу
ISA	Международная стандартная атмосфера
ISIS	Интегрированная резервная приборная система
ISP	Интегрированная коммутационная панель
ISR	Подпрограмма обслуживания прерывания
ITT	Температура между ступенями турбины
IVSI	Индикатор мгновенной вертикальной скорости
JAA	Объединенные авиационные администрации
JTAG	Объединенная группа тестовых действий
JTIDS	Совместная тактическая система распределения информации
KB	Килобайт
KIAS	Узлы приборной скорости
KT	Узел – морская миля в час
KTAS	Узлы истинной воздушной скорости
LAAS	Локальная система функционального дополнения
LADGPS	Локальная дифференциальная глобальная система позиционирования
LCD	Жидкокристаллический дисплей
LDGPS	Локальная дифференциальная глобальная спутниковая система позиционирования
LED	Светоизлучающий диод
LMM	Средний приводной маркерный радиомаяк
LOA	Письмо-соглашение / письмо-разрешение
LOC	Курсовой радиомаяк
LODA	Письмо об одобрении конструкции
LOI	Уровень участия
LOM	Дальний приводной радиомаяк
LORAN	Дальняя радионавигация
LPR	Ротор низкого давления
LPV	Характеристики курсового радиомаяка с вертикальным наведением
LRU	Линейно-заменяемый блок
LSP	Принцип подстановки Лисков
LTE	Потеря эффективности рулевого винта (вертолеты)
MA	Анализ Маркова
MAP	Абсолютное давление во впускном коллекторе или точка прерванного захода
MAPS	Минимальные авиационные эксплуатационные стандарты

Сокращение	Определение
MASPS	Минимальный стандарт авиационных системных характеристик
MB	Мегабайт
MB	Маркерный радиомаяк
MBD	Разработка на основе моделей
MCBF	Среднее число циклов между отказами
MCDU	Многофункциональный блок управления и индикации
MCP	Многоядерный процессор или многоядерная обработка
MC/DC	Модифицированное покрытие условий/решений
MDA	Минимальная высота снижения
MEL	Минимальный перечень оборудования
MF	Средняя частота
MFD	Многофункциональный дисплей
MFDS	Система многофункциональных дисплеев
MIC	Микрофон
MIDO	Окружное управление производственной инспекции
MIDS	Многофункциональная система распределения информации
MILSPEC	Военная спецификация
MKP	Многофункциональная клавиатура
MKR	Маркерный радиомаяк
MLS	Микроволновая система посадки
MM	Средний маркер
MMD	Подвижная карта
MMEL	Главный минимальный перечень оборудования
MNPS	Минимальные навигационные эксплуатационные спецификации
MOA	Зона военных операций
Mode A	Ответчик: передача кода импульсами
Mode C	Ответчик: передача кода и высоты
Mode S	Ответчик: передача кода, высоты и отчетов системы предупреждения столкновений
MOPS	Минимальный стандарт эксплуатационных характеристик
MOSArt	Модульная открытая системная архитектура
MPS	Минимальный стандарт характеристик
MSA	Минимальная безопасная высота
MSG	Сообщение
MSP	Протокол, специфичный для режима S

Сокращение	Определение
MSSS	Службы, специфичные для режима S
MTBF	Средняя наработка на отказ
MTBF	Средняя наработка на отказ
MTBUR	Среднее время между внеплановыми снятиями
MTTF	Среднее время до отказа
MVA	Минимальная векторная высота
MVFR	Предельные визуальные правила полета
NA	Неприменимо
NACO	Национальное управление аэронавигационных карт
NAS	Национальная система воздушного пространства США
NASA	Национальное управление по аэронавтике и исследованию космического пространства
NAV	Навигационный приемник
NAVAID	Навигационное средство
NAVCOMM	Навигационное и связное оборудование или приемник
NAVSTAR-GPS	Спутниковая система навигации, времени и дальномерных измерений
NCATT	Национальный центр подготовки авиационных техников
ND	Навигационный дисплей
NDB	Ненаправленный радиомаяк
NFF	Неисправность не обнаружена
NFPO	Национальное управление летных процедур
NIMA	Национальное агентство изображений и картографии
NM или NMI	Морская миля
NOAA	Национальное управление океанических и атмосферных исследований
NoTAM	Извещение летному составу
NPA	Неточный заход на посадку
NPRM	Уведомление о предлагаемом нормотворчестве
NTAP	Публикация извещений летному составу
NTSB	Национальный совет по безопасности на транспорте
NVD	Прибор ночного видения
NVG	Очки ночного видения
NWS	Национальная метеорологическая служба
OAT	Температура наружного воздуха
OBS	Селектор всенаправленного пеленга
OCC	Центр оперативного управления

Сокращение	Определение
ODA	Полномочие назначенной организации
OEM	Производитель оригинального оборудования
OM	Дальний маркер
OPR	Открытое сообщение о проблеме
OO	Объектно-ориентированный
OOT	Объектно-ориентированная технология
OOTIA	Объектно-ориентированная технология в авиации
OS	Операционная система
OWE/OEW	Эксплуатационная масса пустого воздушного судна/эксплуатационная пустая масса
PA	Гарантия процесса
PA	Система громкой связи
PAL	Пилотное включение светосигнального оборудования
PAPI	Индикатор глиссады точного захода на посадку
PAR	Радиолокатор точного захода на посадку
PASA	Предварительная оценка безопасности воздушного судна
PC	Персональный компьютер
PCL	Управление светосигнальным оборудованием пилотом
PCN	Извещение об изменении изделия
P-Code	Точный код глобальной системы позиционирования
PD	Профильное снижение
PDI	Компоненты параметрических данных
PDL	Руководитель разработки изделия
PDOP	Позиционный фактор ухудшения точности
PDR	Предварительный анализ проекта
PESA	Пассивная электронно-сканируемая решетка
PFD	Основной пилотажный дисплей или основной командный пилотажный прибор
PFDE	Прогнозируемое обнаружение и исключение отказов
PHAC	План сертификации аппаратуры
PLD	Программируемое логическое устройство
PMA	Одобрение изготовителя частей
PMG	Генератор с постоянными магнитами
PND	Основной навигационный дисплей
PNR	Пассивное снижение шума
POA	Одобрение производственной организации

Сокращение	Определение
POF	Этап полета
POH	Руководство по летной эксплуатации для пилота
POS	Положение
PR	Сообщение о проблеме (также может означать “запрос на изменение”)
PRA	Предварительно записанное объявление
PRA	Анализ специфического риска
P-RNAV	Точная зональная навигация
PSAA	План по программным аспектам одобрения
PSAC	План сертификации ПО
PSCP	План сертификации конкретного проекта
PSP	План партнерства в интересах безопасности
PSR	Первичный обзорный радиолокатор
PSSA	Предварительная оценка безопасности системы
PSU	Пассажирский сервисный блок
PTN	Номер отслеживания проблемы
PTR	Сообщение о неисправности программы
PTT	Нажми и говори
QA	Гарантия качества
QAR	Самописец быстрого доступа
QM	Управление качеством
QNH	Барометрическое давление, приведенное к уровню моря
QRH	Краткое справочное руководство
RA	Рекомендация по разрешению конфликта (системы предупреждения столкновений)
RA	Рекомендация по разрешению конфликта (функция воздушного движения)
Rad Alt	Радиовысота
RAI	Индикатор радиовысотомера
RAIM	Автономный контроль целостности приемником
RALT	Радиолокатор или радиовысотомер
RAM	Оперативная память с произвольным доступом
RAT	Воздушная турбина аварийного привода
RCR	Реле обратного тока
RCVR	Приемник
RDMI	Радиомагнитный индикатор дальности
RDP	Система обработки радиолокационных данных

Сокращение	Определение
RDR	Радиолокатор
RDU	Удаленный блок индикации
REF	Опорный/справочный
REIL	Огни идентификации конца ВПП
REL	Относительный
RF	Радиочастота
RFI	Радиочастотные помехи
RHSM	Уменьшенный минимум горизонтального эшелонирования
RJ	Региональный реактивный самолет
RLG	Кольцевой лазерный гироскоп
RLY	Реле

Сокращение	Определение
RMI	Радиомагнитный индикатор
R-NAV	Зональная навигация
RNG	Дальность
RNP	Требуемые навигационные характеристики
ROC	Скорость набора высоты
ROD	Скорость снижения
ROM	Постоянное запоминающее устройство, ПЗУ
RPA	Дистанционно пилотируемое воздушное судно, ДПВС (беспилотный летательный аппарат)
RPM	Обороты в минуту
RSP	Панель переключения в резервный режим
RTE	Маршрут
RTL	Библиотека времени выполнения
RTOS	Операционная система реального времени, ОСПВ
RVR	Дальность видимости на ВПП
RVSM	Сокращенный минимум вертикального эшелонирования
RX	Приемник
SAAR	Специальные требования к воздушному судну и летному экипажу
SAE	SAE
SAR	Поиск и спасание; интеллектуальный регистратор системы мониторинга технического состояния воздушного судна
SAS	Итоговый отчет по ПО
SAT	Статическая температура воздуха
SATCOM	Спутниковая связь
SATNAV	Спутниковая навигация
SATNAV	Спутниковая навигация
SC	Категория управления системой
SCI	Указатель конфигурации ПО (или единица конфигурации ПО)
SCM	Управление конфигурацией ПО
SCMP	План управления конфигурацией ПО
SCR	Рассмотрение соответствия ПО
SCS	Стандарт кодирования ПО
SCS	Блок управления системой
SD	Формат Secure Digital
SDD	Описание проекта ПО

Сокращение	Определение
SDF	Упрощенное направляющее радиосредство
SDP	План разработки ПО
SDRL	Поставляемые материалы поставщика
SDS	Стандарт проектирования ПО
SECI	Указатель конфигурации среды жизненного цикла ПО
SELCAL	Селективный вызов
SES	Спецификация оборудования поставщика
SID	Стандартная схема вылета по приборам
SIU	Блок спутникового интерфейса
SLA	Средство загрузки ПО
SMS	Служба коротких сообщений
SNR	Отношение сигнал/шум
SOI	Этап взаимодействия с сертифицирующим органом
SOP	Стандартная эксплуатационная процедура
SOUP	ПО неизвестного происхождения (иногда употребляется в шутку...)
SOW	Техническое задание
SPP	План программы безопасности
SPR	Рассмотрение планирования ПО
SQA	Гарантия качества ПО
SQAP	План гарантии качества ПО
SQAR	Представитель по гарантии качества ПО
SRATS	Системные требования, распределенные на ПО
SRD	Данные требований к ПО
SRS	Стандарты требований к ПО
SRS	Система опорной скорости или стандарт требований к ПО
SSA	Оценка безопасности системы
SSCV/DR	Твердотельный речевой регистратор кабины экипажа/регистратор данных
SSCVR	Твердотельный речевой регистратор кабины экипажа
SSFDR	Твердотельный регистратор полетных данных
SSPP	План программы безопасности системы
SSR	Вторичный обзорный радиолокатор
SSR	Рассмотрение системной спецификации; альтернативно: рассмотрение спецификации ПО
SSS	Спецификация сегмента системы
STA	Статический временной анализ

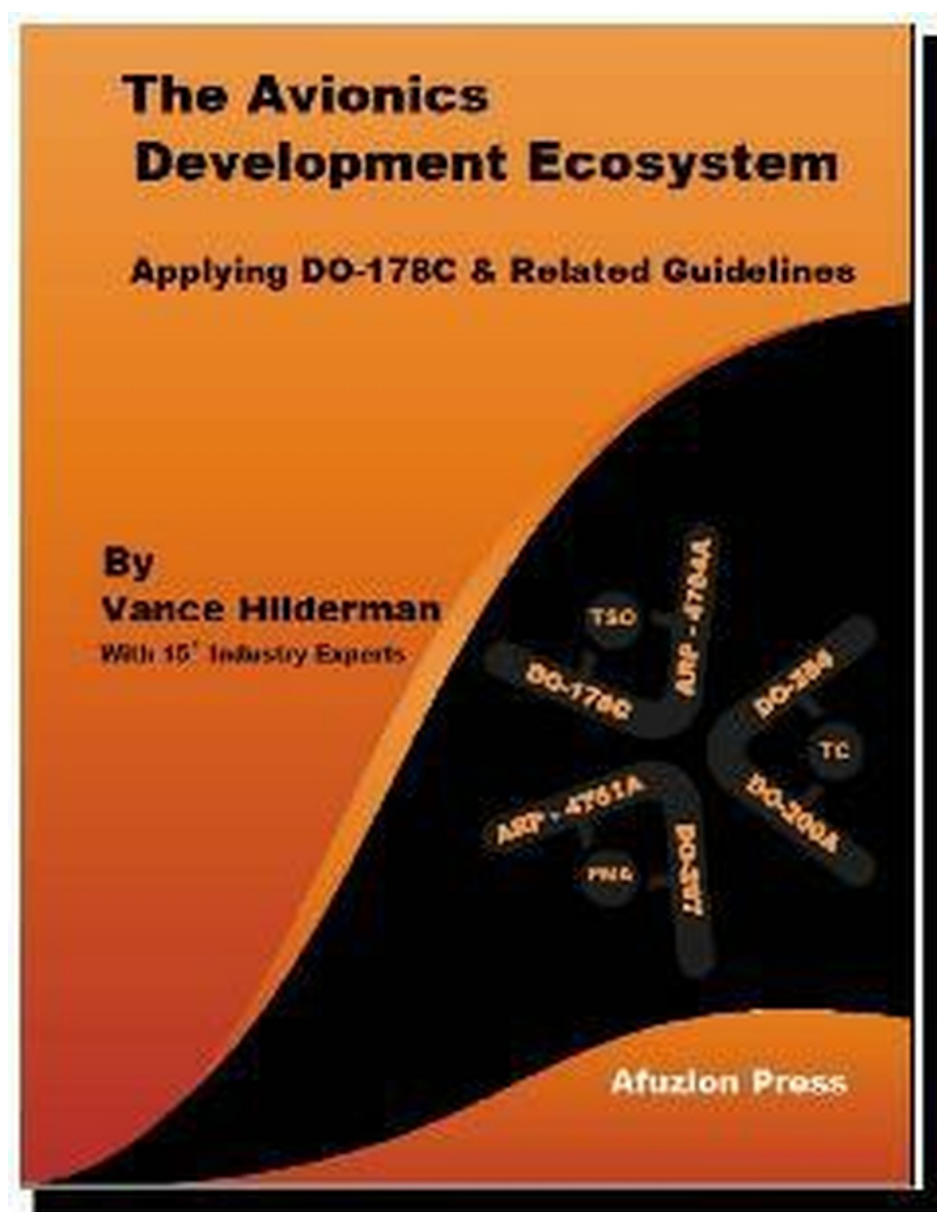
Сокращение	Определение
STAR	Стандартные схемы прибытия в терминальную зону
STAR	Стандартная схема прибытия в терминальную зону
STARS	Стандартная система автоматизации терминальной зоны нового поколения
STC	Дополнительный сертификат типа
STCA	Краткосрочное предупреждение о конфликтной ситуации
STP	Протокол тестирования ПО
STP	Стандартные температура и давление
STS	Спецификация тестирования ПО
SUA	Воздушное пространство специального использования
SVA	Подтверждение валидации ПО
SVCP	Тестовые примеры и процедуры верификации ПО
SVP	План верификации ПО
SVR	Результаты верификации ПО
SW или S/W	ПО
SWCEH	ПО и сложная электронная аппаратура
SYRD	Документ системных требований
T/R	Реверсор тяги или рулевой винт
TA	Консультативное сообщение о воздушном движении
TACAN	Тактическая аэронавигационная система
Tach	Тахометр
TAD	Индикатор осведомленности о рельефе
TAF	Аэродромный прогноз
TAS	Истинная воздушная скорость или итоговый отчет по инструменту
TAT	Истинная температура воздуха или полная температура воздуха
TAWS	Система предупреждения об опасном сближении с землей и информирования о рельефе
TBD	Подлежит определению (или установлению; взаимозаменяемо)
TBO	Время до капитального ремонта или межремонтный ресурс
TC	Сертификат типа или тестовый пример
TCA	Узел управления рычагом тяги или терминальная диспетчерская зона
TCAD	Устройство предупреждения о конфликте воздушного движения
TCAS	Система предупреждения столкновений или система предотвращения столкновений в воздухе
TCF	Минимальная высота безопасного пролета над рельефом
TCI	Указатель конфигурации инструмента
TCN	Тактическая аэронавигационная система

Сокращение	Определение
TCR	Рассмотрение полноты тестирования
TCU	Блок управления тактической аэронавигационной системой
TDOP	Временной коэффициент снижения точности
TDR	Ответчик
TERPS	Процедуры полетов в терминальной зоне и на маршруте
TFR	Временные ограничения полетов
TFT	Тонкопленочный транзистор
TGT	Температура газов перед турбиной или цель
THDG	Истинный курс
TIS	Служба информации о воздушном движении
TK	Угол пути
TKE	Ошибка угла пути
TLA	Угол рычага тяги
TOD	Точка начала снижения
TOR	Эксплуатационные требования к инструменту
TQ	Квалификация инструмента
TQL	Уровень квалификации инструмента
TQP	План квалификации инструмента
TR или T/R	Передачик-приемник или трансивер
TRACON	Диспетчерское радиолокационное управление подходом в терминальной зоне
TRANS	Передача, передавать или переход
TRK	Линия пути
TRP	Ответчик режима S
TRR	Рассмотрение готовности к тестированию / испытаниям
TSO	Технический стандартный приказ
TTL	Настроен на курсовой радиомаяк
TTR	Приемопередатчик TCAS II
TTS	Время до станции
TVE	Полная вертикальная ошибка
TWDL	Двусторонняя линия передачи данных или линия передачи метеоданных терминальной зоны
TWDR	Доплеровский метеорадар терминальной зоны
TWIP	Метеоинформация терминальной зоны для пилотов
TWR	Метеорадиолокатор терминальной зоны
TX	Передача или передавать

Сокращение	Определение
UART	Универсальный асинхронный приемопередатчик
UAS	Беспилотная авиационная система, БАС
UAV	Беспилотный летательный аппарат, БПЛА
UHF	Диапазон ультравысоких частот
ULB	Подводный акустический маяк
UML	Унифицированный язык моделирования
USAF	Военно-воздушные силы США
USB	Универсальная последовательная шина
USGS	Геологическая служба США
UTC	Всемирное координированное время
V	Вольты или напряжение
V&V	Валидация и верификация
V/L	VOR/курсовой радиомаяк
V/NAV	Вертикальная навигация
V/R	Регулятор напряжения
V/REF	Опорная скорость
V/S	Вертикальная скорость
V/TRK	Вертикальная траектория
VASI	Визуальный индикатор глиссады захода
VASIS	Система визуальной индикации глиссады захода
VDF	Пеленгация в диапазоне очень высоких частот
VDL	Линия передачи данных в диапазоне очень высоких частот
VDR	Цифровая радиостанция в диапазоне очень высоких частот
VFO	Генератор переменной частоты
VFR	Правила визуальных полетов
VG/DG	Вертикальный гироскоп / курсовой гироскоп
VGA	Видеографический массив
VHDL	Язык описания аппаратуры VHSIC (сверхскоростных интегральных схем)
VHF	Диапазон очень высоких частот, ОВЧ
VMC	Визуальные метеорологические условия
VMC	Визуальные метеорологические условия
VNE	Скорость, которую запрещено превышать
VNO	Максимальная конструкционная крейсерская скорость
VNR	Навигационный приемник диапазона очень высоких частот

Сокращение	Определение
VOR	Всенаправленный азимутальный радиомаяк диапазона очень высоких частот
VOR/DME	VOR с дальномерным оборудованием
VOR/MB	VOR с маркерным радиомаяком
VORTAC	Комбинация VOR и TACAN
VOX	Голосовая передача
VPA	Заход по вертикальной траектории
VPATH	Вертикальная траектория
VRP	Визуальная контрольная точка
VSI	Индикатор вертикальной скорости
VSM	Минимум вертикального эшелонирования
VSO	Скорость сваливания в посадочной конфигурации
VSWR	Коэффициент стоячей волны по напряжению
VX	Скорость для наилучшего угла набора высоты
VY	Скорость для наилучшей скороподъемности
WAAS	Широкозонная система функционального дополнения
WBS	Структурная декомпозиция работ
WCET	Наихудшее время выполнения
WD	Направление ветра
WINDR	Направление ветра
WMA	Адаптер волновода метеорадиолокатора
WMI	Крепление индикатора метеорадиолокатора
WMS	Широкозонная главная станция
WMSC	Центр коммутации метеосообщений
WMSCR	Замена центра коммутации метеосообщений
WPT	Путевая точка
WRT	Приемопередатчик метеорадиолокатора
WS	Сдвиг ветра
WX	Метеоусловия
WXR	Метеорадиолокатор
WYPT	Путевая точка
XCVR	Трансивер
XFR	Переключение или передача
XMIT	Передавать
XMSN	Передача

Сокращение	Определение
XMTR	Передатчик
XPDR	Ответчик
XTK	Боковое отклонение от линии пути
ZSA	Зональный анализ безопасности



Приложение В: Контрольный перечень требований к авиационной системе

Краткое описание

“Контрольный перечень данных системных требований” помогает убедиться, что требования уровня системы, включая требования безопасности и производные требования, определены надлежащим образом. Для проекта должен быть установлен стандарт системных требований, а также должно быть обеспечено соответствие руководствам SAE ARP4754A и SAE ARP4761/ARP4761A. Авиационные компании-разработчики либо покупают такие контрольные перечни, либо создают собственные, охватывающие все аспекты жизненного цикла авиационной разработки. Распространенный способ найти коммерчески доступные контрольные перечни – ввести в поисковой системе запрос “ARP4754A Requirements Checklists” и просмотреть результаты.

Исходные сведения

В авионических системах требования служат основой разработки. Они должны быть корректными, реализуемыми, однозначно идентифицированными, трассируемыми, декомпозируемыми в требования к аппаратуре и ПО, а также верифицируемыми и валидируемыми. Этот контрольный перечень следует адаптировать под конкретный проект, чтобы обеспечить соблюдение руководства SAE ARP4754A. Следует учитывать, что определение системных требований тесно связано с определением архитектуры системы и выполняется параллельно с ним. Требования заказчика, внешние требования, требования к условиям внешней среды в соответствии с DO-160 и требования безопасности, связанные с оценкой функциональной опасности (Functional Hazard Assessment, FHA), совместно используются для формирования проекта архитектуры системы, включая мониторинг и резервирование. Затем предложенная архитектура итерационно уточняется вместе с развитием системных требований до тех пор, пока не будут определены все системные требования. Этот процесс показан ниже:

Графические материалы к контрольному перечню требований к авиационной системе

Фиксация требований включает:

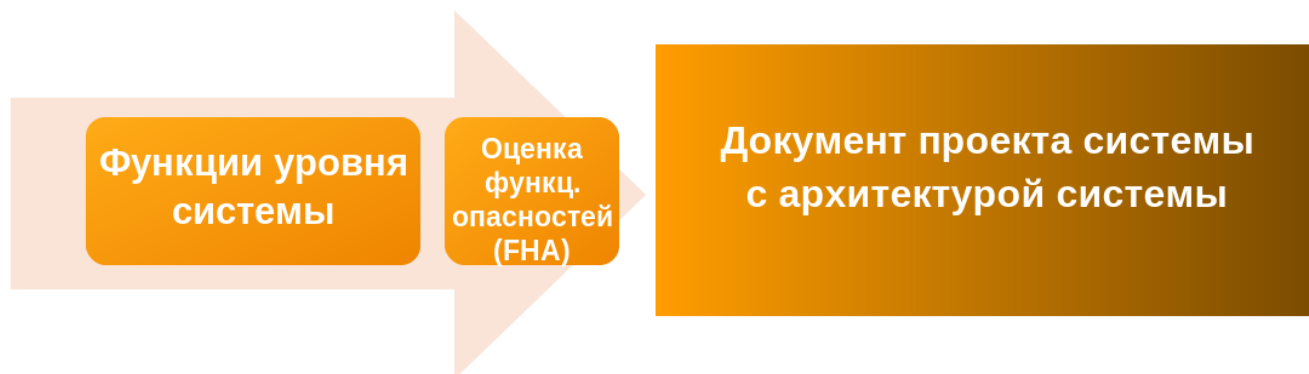
- безопасность;
- производные требования;
- прочие ранее указанные требования.

Примечания:

1. Фиксация требований тесно связана с определением архитектуры системы.
2. Определение архитектуры системы и фиксация требований итеративны до завершения обработки всех требований.

Место фиксации требований в пирамиде деятельности:

1. Сертификация и координация.
2. Гарантия процесса.
3. Управление конфигурацией.
4. Фиксация требований, валидация и верификация (Validation and Verification, V&V).
5. Назначение уровня гарантии разработки (Development Assurance Level, DAL).
6. Безопасность.



Наименование поставщика:	Местоположение:
Наименование системы/линейно-заменяемого блока:	Идентификатор системы/линейно-заменяемого блока:
Рассматриваемый элемент данных (полное наименование артефакта):	
Редакция элемента данных:	Применимые стандарты: SAE ARP4754A, SAE ARP4761/ARP4761A, DO-160, DO-326A (кибербезопасность), если применимо
Дата рассмотрения:	Имя основного рецензента:
Имена дополнительных рецензентов (необязательно):	

Инструкции

Проверьте документ/артефакт на соответствие требованиям, указанным в каждом пункте контрольного перечня. Заполняйте колонку “Да/Нет” следующим образом:

- ” **ДА** “, если артефакт соответствует пункту контрольного перечня.
- ” **НЕТ** “, если артефакт не соответствует пункту контрольного перечня. Добавьте комментарии и/или ссылку на подробные комментарии в колонку “Примечания”.
- ” **Н/П** “, если пункт контрольного перечня неприменим и поэтому не требует рассмотрения. Например, если пункт относится только к уровням А-С, а система относится к уровню D. В колонке “Примечания” укажите обоснование для “Н/П”.

Примечание: все несоответствия должны быть устранены до утверждения документа.

1.0 Общие пункты контрольного перечня

Критерий	Да или нет	Примечания
a. Рассмотрен ли этот документ применимым сертификационным органом проекта-заявителя?	ДА НЕТ	
b. Рассмотрен ли этот документ применимым представителем по гарантии процесса проекта-заявителя?	ДА НЕТ	
c. Находится ли этот артефакт (документ) под управлением конфигурацией, как определено в ссылочном плане управления конфигурацией?	ДА НЕТ	
d. Указаны ли в документе наименование системы, дата выпуска, идентификатор документа и уровень редакции?	ДА НЕТ	
e. Определены ли применимые термины и сокращения для всех сокращений, используемых в документе?	ДА НЕТ	
f. Полон ли раздел ссылочных документов и указаны ли в нем все применимые документы проекта, на которые есть ссылки или которые связаны с сертификационными/инженерными критериями?	ДА НЕТ	

2.0 Специальные пункты контрольного перечня

Вопрос	Соответствует	Примечания
Системные требования		
Распределены ли на систему применимые внешние требования к условиям среды, например из DO-160?	ДА НЕТ Н/П	
Распределены ли на систему применимые внешние требования к характеристикам, например из применимых технических стандартных приказов (Technical Standard Order, TSO)?	ДА НЕТ Н/П	
Включены ли потенциальные отказы и выходные данные процесса FHA в системные требования, связанные с безопасностью?	ДА НЕТ Н/П	
Согласуются ли системные требования с архитектурой уровня системы?	ДА НЕТ Н/П	
Описано ли распределение системных требований на аппаратуру или ПО с учетом требований, связанных с безопасностью, и потенциальных отказных состояний?	ДА НЕТ Н/П	
Учтены ли отказные состояния, выявленные при FHA, через требования безопасности, распределенные на систему?	ДА НЕТ Н/П	
Соответствуют ли системные требования стандарту системных требований и руководству SAE ARP4754A, а для требований безопасности – также SAE ARP4761/ARP4761A?	ДА НЕТ	
Если требование отклоняется от стандарта требований, проводится ли независимая оценка отклонения до его принятия?	ДА НЕТ	
Являются ли системные требования:		
однозначными?	ДА НЕТ	
согласованными?	ДА НЕТ	
полными, без неопределенных условий?	ДА НЕТ	
верифицируемыми анализом или испытанием? Примеры неverified терминов: “гибкий, простой, достаточный, безопасный, специальный, адекватный, поддерживающий, удобный для пользователя, пригодный к использованию, при необходимости, если требуется, подходящий, быстрый, переносимый, облегченный, малый, большой, максимизировать, минимизировать, достаточный, робастный, быстро, легко, ясно, другие расплывчатые наречия, другие неопределенные глаголы действия”?	ДА НЕТ	
сформулированными количественно, с допусками там, где это применимо?	ДА НЕТ	

Вопрос	Соответствует	Примечания
Сформулировано ли требование утвердительно, а не отрицательно, то есть без формы “не должно”?	ДА НЕТ	
Является ли каждое требование уникальным, чтобы исключить полное или частичное дублирование?	ДА НЕТ	
Не описывают ли системные требования детали проектирования аппаратуры/ПО или верификации, за исключением заданных и обоснованных проектных ограничений?	ДА НЕТ	
Не содержат ли системные требования положений, которые лучше отнести к требованиям высокого уровня к ПО и/или требованиям к аппаратуре?	ДА НЕТ	
Определены ли системные требования, которые выявляют и предотвращают опасные состояния системы?	ДА НЕТ Н/П	
Если применимо, описывают ли системные требования, как система использует любой компонент параметрических данных (Parameter Data Item, PDI), являющийся надмножеством конфигурационных данных, включая:		
структуру PDI?	ДА НЕТ Н/П	
атрибуты PDI?	ДА НЕТ Н/П	
если применимо, значения, согласованные со структурой PDI и атрибутами его элементов данных?	ДА НЕТ Н/П	
Определены ли функциональные и эксплуатационные требования для каждого режима работы?	ДА НЕТ Н/П	
Охватывают ли системные требования критерии характеристик, например точность и погрешность?	ДА НЕТ Н/П	
Определены ли ограничения на объем памяти?	ДА НЕТ Н/П	
Определены ли временные требования и ограничения?	ДА НЕТ Н/П	

Охватывают ли системные требования интерфейсы аппаратуры и системы, включая:		
протоколы?	ДА НЕТ Н/П	
форматы?	ДА НЕТ Н/П	
частоту входных данных?	ДА НЕТ Н/П	
частоту выходных данных?	ДА НЕТ Н/П	
Определены ли требования к обнаружению отказов?	ДА НЕТ Н/П	
Определены ли требования к мониторингу безопасности и соответствуют ли они SAE ARP4754A и SAE ARP4761/ARP4761A?	ДА НЕТ Н/П	
Если система включает требования к обособлению (partitioning), рассмотрены ли следующие пункты:		
определены ли требования к обособлению, распределенные на систему?	ДА НЕТ Н/П	
описано ли, как обособленные компоненты системы взаимодействуют друг с другом?	ДА НЕТ Н/П	
определен ли уровень системы для каждого обособленного компонента?	ДА НЕТ Н/П	
Определены ли требования к встроенному контролю и мониторингу, включая обнаружение, локализацию, выдачу сообщений, контроль работоспособности и переключение?	ДА НЕТ Н/П	
Определены ли требования к восстановлению после отказов системы?	ДА НЕТ Н/П	
Определены ли требования к восстановлению после ненормальных условий, включая автоматический перезапуск, отказ питания и переходные процессы?	ДА НЕТ Н/П	
Содержат ли данные системных требований требование о том, чтобы идентификатор единицы конфигурации системы/ПО был встроен в систему/ПО и задан так, чтобы его можно было наблюдать, считать или определить иным способом?	ДА НЕТ Н/П	
Переданы ли производные требования в процесс оценки безопасности системы?	ДА НЕТ Н/П	
Если применимо, полны ли требования к резервированию системы, включая подробные сведения об условиях, которые могут вызвать переключение, временных характеристиках и определении работоспособности, включая голосование, особенно для систем DAL A или DAL B?	ДА НЕТ Н/П	
	ДА НЕТ Н/П	

Охватывают ли системные требования интерфейсы аппаратуры и системы, включая:		
Обеспечивают ли требования независимость функций и их мониторов, требуя, чтобы монитор и его защитный механизм не становились неработоспособными из-за того же отказного состояния, которое вызывает опасность?		
Если система включает требования к данным, модифицируемым пользователем, рассмотрены ли следующие пункты:		
Имеются ли требования, определяющие механизмы, которые не позволяют пользовательской модификации повлиять на безопасность системы независимо от того, корректно ли эти механизмы реализованы?	ДА НЕТ Н/П	
Находится ли средство защиты от пользовательской модификации на том же уровне системы, что и функция, которую оно защищает от ошибок в модифицируемом компоненте?	ДА НЕТ Н/П	
Запрещено ли пользователю модифицировать систему, если для этой модификации не показано соответствие SAE ARP4754A?	ДА НЕТ Н/П	
Предусмотрено ли, что в момент пользовательской модификации пользователь принимает на себя ответственность за все аспекты модифицируемой пользователем системы, например за управление конфигурацией системы, гарантию качества системы и верификацию системы?	ДА НЕТ Н/П	
Если система включает требования к ПО или аппаратуре с выбираемыми опциями, рассмотрен ли следующий пункт:		
Когда в системе предусмотрены программируемые опции или конфигурации, есть ли средства, исключающие непреднамеренный выбор не одобренных конфигураций для целевого вычислителя в среде установки?	ДА НЕТ Н/П	
Если система имеет режим по умолчанию при загрузке неподходящей системы или неподходящих данных, заданы ли для каждого обособленного компонента системы требования к работе в этом режиме, связанные с безопасностью и учитывающие потенциальное отказное состояние?	ДА НЕТ Н/П	
Имеются ли требования к установке и соответствуют ли они оценке безопасности, особенно в части анализа общих причин?	ДА НЕТ Н/П	
Имеются ли требования к эксплуатации и соответствуют ли они оценке безопасности?	ДА НЕТ Н/П	

Охватывают ли системные требования интерфейсы аппаратуры и системы, включая:		
Имеются ли требования к техническому обслуживанию и соответствуют ли они оценке безопасности?	ДА НЕТ Н/П	
Если система включает бортовой механизм отображения, являющийся средством подтверждения соответствия воздушного судна сертифицированной конфигурации, разработана ли эта система по наивысшему уровню среди загружаемых в нее систем, или процесс оценки безопасности системы обосновывает целостность сквозной проверки идентификации конфигурации системы?	ДА НЕТ Н/П	
Если система имеет функцию загрузки, включая вспомогательные системы и процедуры, определено ли средство обнаружения неправильных сочетаний ПО, аппаратуры и/или воздушного судна, и предусмотрена ли защита, соответствующая отказному состоянию этой функции?	ДА НЕТ Н/П	
Системные требования		
Обеспечена ли трассируемость между системными требованиями и требованиями безопасности, чтобы можно было верифицировать полную реализацию системных требований и видеть производные требования, которые напрямую не трассируются к системным требованиям?	ДА НЕТ Н/П	
Можно ли декомпозировать каждое системное требование в требования высокого уровня к аппаратуре или ПО?	ДА НЕТ Н/П	
Можно ли валидировать и верифицировать каждое системное требование одним или несколькими тестовыми примерами?	ДА НЕТ Н/П	

Раздел комментариев: включите дополнительные комментарии по рассмотрению, сводку наблюдений и необходимые действия, требуемые для принятия этого документа.

Дополнительные комментарии по рассмотрению

- a.
- b.
- c.

Сводка наблюдений

- a.
- b.

- с.

Необходимые действия для принятия этого документа

- а.
- b.
- с.